Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung
Fachgebiet Neuroinformatik und Kognitive Robotik

Dissertation

Ein Beitrag zur robusten Nutzerwahrnehmung auf realwelttauglichen Assistenzrobotern in häuslichen Szenarien

zur Erlangung des akademischen Grades Doktoringenieur (Dr.-Ing.)

Dipl.-Inf. Michael Volkhardt

Tag der Einreichung: 17. Januar 2018
Tag der wissenschaftlichen Aussprache: 16. Oktober 2018

Gutachter:
1. Univ.-Prof. Dr.-Ing. Horst-Michael Groß, Technische Universität Ilmenau
2. Univ.-Prof. Dr. sc. techn. Bastian Leibe, RWTH Aachen University
3. Univ.-Prof. Dr.-Ing. habil. Gerhard Rigoll, Technische Universität München

urn:nbn:de:gbv:ilm1-2018000374

Für Marina und Jonas

Kurzfassung

Die höhere Lebenserwartung der Bevölkerung und eine rückläufige Geburtenrate führen zu einem steigenden Anteil älterer Menschen in der Gesellschaft. Mobile Assistenzroboter sollen ältere Personen zukünftig in ihren Wohnungen unterstützen. Um sinnvolle Funktionen und Dienste anbieten zu können, muss der Roboter Personen in seiner Umgebung wahrnehmen können. Das häusliche Szenario stellt dabei aufgrund seiner Komplexität eine Herausforderung für die Erkennungsalgorithmen dar. Komplexität entsteht beispielsweise durch unterschiedliche Einrichtungsmöglichkeiten, schwierige Beleuchtungsbedingungen und variable Nutzerposen.

Die Dissertation stellt eine Architektur zur Personenwahrnehmung für mobile Roboter vor. Die modulare Architektur beschreibt die verwendeten Komponenten und deren Kommunikation untereinander. Aufgrund der Modularität können einzelne Komponenten schnell integriert oder ausgetauscht werden.

Die Arbeit evaluiert eine Vielzahl von multi-modalen Detektionsverfahren auf Basis von Laser-, Kamera- und 3D-Tiefendaten. Ausgewählte Algorithmen werden für Anwendungsszenario angepasst und weiterentwickelt. Die Hypothesen der Detektoren werden durch einen Personentracker raumzeitlich gefiltert und fusioniert. Besonderheiten des Personentrackers umfassen die Unterstützung mehrerer Filter und Systemmodelle, die Integration von nicht unabhängigen und verspäteten Beobachtungen, die Schätzung der Existenzwahrscheinlichkeit sowie die Integration von Umgebungswissen.

Um Nutzer, welche sich nicht im Sichtbereich des Roboters befinden, in der Wohnung zu finden, werden verschiedene Suchverfahren vorgestellt. Das fortschrittlichste Verfahren verwendet eine explorative Suche, um die gesamte Wohnung effektiv zu durchsuchen. Dabei werden falsch-positiv Detektionen ausgeschlossen und mit dynamischen Hindernissen und nicht erreichbaren Räumen umgegangen.

Die Arbeit stellt ein Verfahren für die Erkennung von gestürzten, am Boden liegenden Personen vor. Die auf Tiefendaten basierende Erkennung erlaubt es dem Roboter, Personen von anderen Objekten oder Tieren in der Wohnung zu unterscheiden.

Die entwickelten Algorithmen wurden im realen Anwendungsszenario evaluiert, indem der Roboter für bis zu 3 Tage in den Wohnungen von Senioren zur freien Nutzung verblieb. Die Experimente zeigten, dass die vorgestellte Architektur zur Personenwahrnehmung robust genug arbeitet, damit der Roboter mithilfe seiner Dienste einen Mehrwert für die Senioren liefern kann.

Abstract

The increased life expectancy of the population and declining birth rates lead to an increasing proportion of elderly people in the modern society and hence an increasing need for age care. Mobile robots can assist users in theirs homes by means of services and companionship. To provide useful functionalities, the robot must be able to observe the user in the environment. The domestic scenario poses a challenge for people detection and tracking algorithms through its complexity caused, among others, by variable furnishing options, difficult lighting conditions and various user poses.

This thesis presents an architecture for people detection and tracking for mobile robots in domestic environments. The modular architecture describes the used components and their communication with each other. Due to the modularity of design, the individual components can be easily integrated or exchanged.

This work evaluates a variety of multi-modal detection methods based on laser data, camera data and 3D depth data. Suitable algorithms are being applied, adapted and enhanced. The detections are processed by a person tracker to allow for spatial-temporal filtering. Important features of the person tracker include the support of multiple filters and system models, the integration of coupled observations and out-of-sequence measurements, the estimation of the existence probability and the integration of environmental knowledge.

The thesis proposes various methods to search and locate users in the apartment, which have left the robots limited field-of-view. The most advanced method uses an exploratory search method to examine the environment effectively. It handles false positive detections, dynamic obstacles and inaccessible rooms in a reasonable manner.

Furthermore, this work presents a method to detect people that have fallen to the ground given occlusion. The method uses the depth data of a Kinect sensor mounted on the mobile robot. Point clouds are segmented, layered and classified to distinguish fallen people from furniture, household objects and animals.

The developed algorithms were evaluated in a real-world scenario, by allowing the robot to stay in retirement homes for up to three days. The experiments showed that the presented architecture for people detection and tracking is robust enough, so that the robot's services proved to provide an added value to the seniors citizens.

Danksagung

Mein herzlicher Dank gilt all denen, die mich auf unterschiedliche Weise bei der Erstellung dieser Dissertation unterstützt haben.
Zuerst möchte ich meinem Doktorvater Prof. Dr. Horst-Michael Groß danken, der mir immer unterstützend zur Seite stand und mir die Arbeit am Fachgebiet Neuroinformatik und Kognitive Robotik ermöglicht hat.
Ich danke allen Mitarbeitern des Fachgebiets Neuroinformatik und Kognitive Robotik für eine großartige Zeit mit vielen schönen Erinnerungen. Insbesondere möchte ich mich bei Ronny Stricker, Christian Vollmer, Christof Schröter, Steffen Müller, Erik Einhorn und Christoph Weinrich für die vielen fachlichen Diskussionen und freundschaftlichen Gespräche bedanken.
Weiterhin möchte ich den Studenten danken, die durch ihre Abschlussarbeiten einen Teil zu dieser Dissertation beigetragen haben.
Ich möchte meiner Familie für ihre Liebe und Unterstützung danken. Insbesondere meiner Frau, die mich stets motiviert hat, die Arbeit fertig zu stellen.

Inhaltsverzeichnis

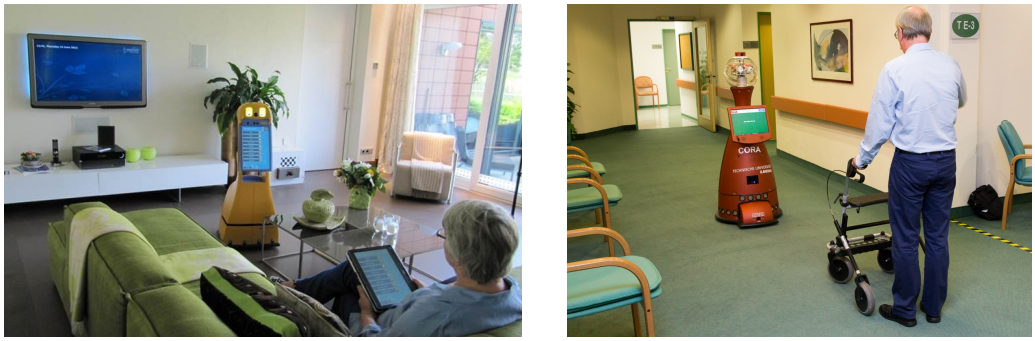
1. Einleitung

Seit jeher entwickelt der Mensch stetig neue Technologien, um sein Leben angenehmer zu gestalten. In der heutigen Zeit umgibt uns eine Vielzahl technologischer Errungenschaften, die von den meisten Menschen als selbstverständlich wahrgenommen werden. Beispielsweise ermöglichen eine umfassende Grundversorgung und hohe medizinische Standards, der Weltbevölkerung immer älter zu werden. Neben unzähligen Vorteilen, wie einer höheren Lebensqualität, ergeben sich daraus allerdings auch Probleme. Die gestiegene Lebenserwartung hat zur Folge, dass ein höherer Pflegebedarf entsteht, der aufgrund des demografischen Wandels nicht durch die nachfolgende Bevölkerungsschicht gedeckt werden kann. Aus dem Mangel an Pflegepersonal ergeben sich erhöhte Kosten, verminderte Qualität sowie geringere Zeit für soziale Kontakte während der Pflege. Der Wunsch älterer Personen, möglichst lange unabhängig zu Hause leben zu können, entwickelt sich daher zu einer großen Herausforderung des Gesundheits- und Pflegesektors (Meyer 2011). Diesen Wunsch versucht man unter anderem durch sogenannte *ambient assisted living*-Systeme (AAL) zu erfüllen.
AAL-Systeme sind umgebende Methoden oder Technologien, die Menschen in ihrer häuslichen Umgebung nutzerzentriert unterstützen, um ihre Gesundheit, ihre Sicherheit und ihr Wohlbefinden zu verbessern. Die Akzeptanz solcher Systeme hängt maßgeblich vom Angebot an nützlichen Diensten ab. Dienste wie Erinnerungen an Termine, Wasser- und Medikamenteneinnahme, Sicherheitsfunktionen und Sturzerkennung, Unterhaltung und soziale Interaktion gehören zu den meistgenannten Wünschen von Senioren an intelligente, unterstützende Systeme (Huijnen u. a. 2011). Die ausreichende Flüssigkeitszufuhr wird von älteren Personen häufig vernachlässigt und stellt einen Grund dar, warum sich Personen in Pflege begeben müssen. Weiterhin zeigen Statistiken, dass für ältere Personen erhöhte Gefahr eines Sturzes besteht (Lord u. a. 2003). Die für den sozialen Kontakt mit Angehörigen und Freunden häufig angewandte Videotelefonie kann auch zur Telepräsenz und Telemedizin verwendet werden. Weitere Funktionalitäten umfassen die Messung der Vitalparameter, kognitive Übungen (Gehirnjogging) und verschiedene einfache Bewegungsübungen. In AAL-Lösungen werden typischerweise mehrere Sensoren und Aktuatoren in einem Gesamtsystem kombiniert. Dabei können sowohl statische Systeme (*smart-home* Technologie) als auch mobile Systeme (*Assistenzroboter*) zum

Einsatz kommen.
Mobile Assistenzroboter bieten dabei mehrere Vorteile. Einerseits ermöglichen sie die Umsetzung aller oben genannten Dienste, andererseits erlaubt die Mobilität des Roboters, Service dort anzubieten, wo er benötigt wird. Dienste wie Videokonferenz, Telepräsenz und die Erkennung und Reaktion auf Stürze werden durch die Bewegungsfreiheit des Roboters erleichtert. Weiterhin erlaubt die Verkörperung (engl. *embodiment*) des Roboters eine natürlichere Mensch-Maschine-Interaktion und fördert die Akzeptanz von proaktivem Verhalten des Systems, wie beispielsweise das Erinnern an Medikamenteneinnahme oder das Auffordern zum kognitiven Training. Durch die Vermenschlichung wird das System nicht mehr als Technik wahrgenommen, sondern als umgebender Begleiter im Alltag. Ein sozialer Partner kann langfristig besser motivieren und unterstützen als beispielsweise ein TV-Gerät oder ein Tablet. Die Autonomie des Roboters, wie das selbstständige Aufladen der Akkus oder die Navigation durch die Wohnung, erhöht den Nutzeffekt zusätzlich (Robinson u. a. 2014).
Der Traum von hochintelligenten, menschenähnlichen Robotern liegt noch in der Zukunft. Vor allem die Lösung vieler Problematiken, wie der menschliche Gang, die Manipulation mit Armen sowie die Nachbildung der komplexen kognitiven Fähigkeiten des Menschen befinden sich noch in den Kinderschuhen. Andererseits wagen einige Projekte bereits den Schritt aus dem Labor in reale Wohnungen, um die Fähigkeiten von Assistenzrobotern an Endnutzern zu untersuchen.
Dabei müssen Assistenzroboterroboter in der Lage sein, vorhandene Personen in ihrer Umgebung wahrzunehmen. Die Detektion und Lokalisation von Personen mithilfe der Sensorik ist eine unabdingbare Voraussetzung für eine natürliche Mensch-Maschine-Kommunikation und nahezu alle der genannten Serviceleistungen. Auch das Suchen und Auffinden einer Person in der Wohnung ist von entscheidender Bedeutung, da der Erfassungsbereich der Robotersensorik beschränkt ist und der Nutzer diesen oft verlässt, indem er beispielsweise den Raum wechselt. Der Hauptteil dieser Dissertation befasst sich daher mit der Wahrnehmung und der Suche von Personen in häuslichen Umgebungen. Im folgenden Abschnitt wird ein Überblick über den Fokus der Dissertation und deren Methoden gegeben.

1.1. Inhalt und Anspruch der Arbeit

Die Dissertation entstand im Rahmen der Projekte CompanionAble und SERROGA, welche sich mit der Entwicklung eines Serviceroboters für die häusliche Umgebung beschäftigen (CompanionAble 2008; SERROGA 2012). Der Schwerpunkt der Arbeit liegt in der Entwicklung einer Architektur für mobile



(a) CompanionAble/SERROGA **(b)** ROREAS

Abbildung 1.1.: Roboterprototypen der Projekte CompanionAble/SERROGA und ROREAS, welche die entwickelte Architektur zum Personentracking verwenden.

Roboter, um Personen in häuslichen Umgebungen wahrzunehmen. Wahrnehmung umfasst hierbei das Suchen von Personen in der Wohnung, die Detektion und Lokalisation sowie das raumzeitliche Verfolgen von Personenpositionen. Personenwahrnehmung, speziell von Fußgängern, ist seit geraumer Zeit Gegenstand intensiver Forschungen. Das häusliche Szenario unterscheidet sich jedoch von Labor- oder Außenanwendungen in vielerlei Hinsicht. Eine relativ kleine mobile Plattform, enge Räume, wechselnde Lichtverhältnisse, Verdeckungen und eine Vielzahl an unterschiedlichen Nutzerposen sind nur einige Aspekte, die das Szenario anspruchsvoll werden lassen und den Rahmen der Dissertation vorgeben.

Die zum Einsatz kommende Roboterplattform (Abbildung 1.1(a)) reiht sich in die Liste der sozialen Assistenzroboter ein (Tapus u. a. 2007; Gross u. a. 2012). Der Roboter kann den Nutzer hauptsächlich mittels Kamera und Laserscanner erfassen. Die in der Arbeit vorgestellten Methoden orientieren sich an der vorhandenen Sensorik und der verfügbaren Rechenleistung des Roboters sowie den Zielen der Rahmenprojekte.

Die zu entwickelte Architektur zur Nutzerwahrnehmung soll jedoch allgemeingültig, roboterunabhängig und auf andere Szenarien übertragbar sein. Als vorgezogener Beleg wird der entwickelte Personentracker und Teile der Personendetektoren erfolgreich auf einem anderen Roboter im ROREAS Projekt eingesetzt (ROREAS 2013) (Abbildung 1.1(b)). Neben dem Personentracking werden neue Methoden vorgestellt, die es ermöglichen sollen, Personen in einer häuslichen Umgebung effizient zu suchen, sowie Stürze zu erkennen.

Die wesentlichen im Rahmen dieser Arbeit zu erbringenden methodisch-technischen Beiträge der Arbeit lassen sich folgendermaßen zusammenfassen:

- **Die Systemarchitektur** beschreibt die Gesamtapplikation zur Nutzerwahrnehmung. Die Architektur ermöglicht das Zusammenspiel von Personendetektoren und -tracker sowie der darauf aufbauenden Personensuche. Zusätzlich integriert die Architektur Algorithmen zum Folgen einer Person sowie der Erkennung von Stürzen und stellt eine Verbindung zu anderen Modulen her. Sie soll größtenteils allgemeingültig und auf andere Szenarien und Roboter übertragbar sein. Da der Assistenzroboter aktiv mit dem Nutzer interagiert, müssen alle Anwendungen und Algorithmen in Echtzeit auf der Hardware der Plattform ausgeführt werden können. Als Motivation für die Ressourceneffizienz der eingesetzten Methoden dient die Hypothese, dass nicht nur der eingesetzte Roboter, sondern alle mobilen Assistenzroboter der nächsten Jahre in einem vertretbaren Preissegment ähnlichen Limitierungen ausgesetzt sind. Die zu entwickelten Methoden können daher Markteintrittsbarrieren von Assistenzrobotern bezüglich Sensor- und PC-Ausstattung senken. Die Arbeit stellt eine neuartige modulare Systemarchitektur vor, welche die Erweiterbarkeit und Übertragbarkeit erhöht. Ebenfalls stellt das Gesamtkonzept aus Detektoren, Tracker und Personensuche in dieser Form einen Neuheitswert dar.

- **Personendetektion** umfasst die Wahrnehmung und Lokalisation von Personen und ist Gegenstand intensiver Forschungen. Aktuelle Verfahren erreichen eine beeindruckende Qualität und Geschwindigkeit, sind jedoch noch an Randbedingungen geknüpft, die sich im betrachteten Szenario nicht einhalten lassen. Die Dissertation stellt daher neben passenden Detektionsalgorithmen auch Anpassungen etablierter Verfahren vor, um den Einsatz auf der mobilen Plattform zu ermöglichen. Auf Basis von Laserdaten und Kamerabildern werden Personen multi-modal über Beinpaare, Gesicht, Oberkörper- und Körpersilhouette detektiert. Zum Vergleich werden alternative Verfahren aus anderen Szenarien (z. B. Fußgängerdetektion) und anderen Architekturvoraussetzungen (Stereo- oder Wärmebildkamera, GPU-Algorithmen, oder Offline-Verfahren) aufgeführt. Die methodisch-technischen Innovationen umfassen die Entwicklung, Auswahl und Anpassung von Detektionsverfahren für die Systemarchitektur. Ein Alleinstellungsmerkmal ist die konsequente Modellierung von Personendetektionen in 3D-Weltkoordinaten und die Verlagerung des Sensormodells in die Detektionsmodule.

- **Personentracking** umfasst das raumzeitliche Verfolgen von Personenpositionen. Trackingalgorithmen kombinieren die verrauschte Information unterschiedlicher Detektionsverfahren und erzielen somit ein robuste-

res Gesamtergebnis. In der Literatur existiert eine Vielzahl an Algorithmen, wobei speziell Bayes-Filter eine entscheidende Rolle in Echtzeitanwendungen spielen. In der Dissertation soll ein allgemeines Framework geschaffen werden, das die Implementierung verschiedener Filteralgorithmen und Systemmodelle ermöglicht. Weiterhin soll ein Wechsel dieser Filter und Modelle zur Laufzeit möglich sein, um die verschiedenen Verhaltensweisen von Personen zu modellieren. Die Dissertation adressiert zusätzlich die Existenzschätzung von Hypothesen, die Behandlung korrelierter Mehrfachdetektionen und veralteter Beobachtungen, sogenannter *out of sequence measurements*. Die Leistung des Trackers wird maßgeblich durch die eingesetzten Detektionsmodule beeinflusst. Aufgrund der oben genannten Einschränkungen ist nicht in jeder Situation, z. B. unter variablen Posen, mit einem perfekten Trackingergebnis zu rechnen. Dennoch stellt der entwickelte Personentracker mit den eingesetzten Detektoren ein realwelttaugliches System auf einem mobilen Roboter dar, welches in dieser Form bisher nicht existiert.

- **Intelligente Nutzersuche.** Aufgrund des eingeschränkten Wahrnehmungsbereichs der Robotersensorik kann der Nutzer nicht zu jeder Zeit erfasst werden. Beispielsweise kann sich die Person in einen anderen Raum bewegen, ohne zu wünschen, dass ihr der Roboter folgt. Stehen Ereignisse an, die eine Interaktion mit dem Nutzer erforderlich machen, muss der Roboter die Person in der Wohnung suchen und finden können. Hierbei spielen Robustheit und Geschwindigkeit eine entscheidende Rolle. In der Dissertation werden drei neuartige Methoden zur Lokalisation des Nutzers in einer häuslichen Umgebung vorgestellt:

  1. Visuelle Detektion von Personen an Plätzen mit hoher Aufenthaltswahrscheinlichkeit: Hierbei werden die visuellen Eigenschaften von unbesetzten Plätzen in der Wohnung gelernt. Mithilfe eines visuellen Personenmodells kann ein Klassifikator anschließend entscheiden, ob der Platz belegt ist oder nicht.

  2. Verifikation von Hypothesen und Ausschließen von falsch-positiv Detektionen: Der vorhandene Personentracker wird durch Module verbessert, die sich nicht im parallelen Betrieb einsetzen lassen und daher bei Bedarf aufgerufen werden. Die Suchstrategie des Roboters wird mit Aufenthaltswahrscheinlichkeitskarten beschleunigt.

  3. Explorative Suche: Der Roboter führt eine natürliche Suche in der Wohnung durch, indem an Stellen gesucht wird, an denen der Nutzer vermutet wird, beziehungsweise bisher noch nicht gesucht wurde.

- **Sturzerkennung.** Kommerziell verfügbare Produkte zur Sturzerkennung liefern bisher noch keine vollständig zufriedenstellenden Ergebnisse. Daher wurde im Rahmen der Dissertation ein Verfahren entwickelt, das gestürzte Personen mithilfe eines mobilen Roboters erkennt. Als Herausforderung kann der Sturz außerhalb des Sichtbereichs der Robotersensorik auftreten. Mittels verschiedener Form- und Oberflächenfeatures, die aus Tiefenkameradaten extrahiert werden, können am Boden liegende Personen von anderen Objekten unterschieden werden. Die Nutzung von Tiefendaten erlaubt Robustheit gegenüber Beleuchtungsschwankungen und erleichtert die Erfassung der hohen Varianz an unterschiedlichen Posen gestürzter Personen. Das Verfahren ist, nach Kenntnis des Autors, das erste, welches gestürzte Personen auf einer mobilen Plattform mittels Tiefendaten erkennt.

- **Einsatz und Evaluation im häuslichen Szenario.** Die Roboterplattform wird im Projekt SERROGA in realen Wohnungen evaluiert. Dabei verbleibt der Roboter in den Wohnungen der Nutzer, welche im Verlauf der Untersuchungen verschiedenste Services des Roboters nutzen, z. B. Videotelefonie mit automatischer Suche bei eingehenden Anrufen, Terminerinnerung und „Folge mir“. Dabei wird das entwickelte System zur Nutzerwahrnehmung verwendet. Der Einsatz mobiler Roboter in häuslichen Umgebungen im täglichen Gebrauch ist zurzeit noch relativ unerforscht. Die Arbeit untersucht daher die Performance und Robustheit des Systems während der Nutzertests und bewertet das Zusammenspiel der Module in der Gesamtapplikation. Zuvor werden zur Ermittlung quantitativer Ergebnisse und zwecks Reproduzierbarkeit Experimente in realistischen, aber nachgestellten, häuslichen Umgebungen durchgeführt. Während sich reale Wohnungen für qualitative Aussagen eignen, stellt es sich als schwierig heraus, bestimmte Funktionalitäten (z. B. Trackingperformance oder Sturzerkennung) mit Senioren quantitativ zu untersuchen. Die Dissertation liefert einen ingenieurtechnischen Beitrag auf das Anwendungsfeld, indem gezeigt wird, welche Probleme durch die vorgestellten Methoden im häuslichen Szenario gelöst werden können.

Die Dissertation setzt sich somit zum Ziel, Methoden zu entwickeln, die einen wesentlichen Anteil an einer realwelttauglichen Gesamtapplikation für mobile Assistenzroboter im häuslichen Umfeld ausmachen.

1.2. Gliederung und Leseleitfaden

Die Arbeit gliedert sich in sechs Kapitel. Zunächst werden in Kapitel 2 das häusliche Szenario und dessen Besonderheiten sowie der Roboter und dessen Sensorik beschrieben. Weiterhin werden kurz verwandte Arbeiten und aktuelle Projekte der Assistenzrobotik vorgestellt. Aus Szenario und eingesetztem Roboter leitet sich die Systemarchitektur ab. Die Architektur vereint alle Algorithmen dieser Dissertation und stellt einen Bezug zu weiteren Modulen des Roboters her. Kapitel 2 spannt daher den Bogen für die nachfolgenden Kapitel. Anschließend gibt Kapitel 3 einen Überblick über die eingesetzten Personendetektoren sowie möglichen Alternativen. Kapitel 4 stellt das entwickelte Tracking-Framework vor. Kapitel 3 und 4 können unabhängig voneinander gelesen werden, wobei die Hypothesen der vorgestellten Personendetektoren als Eingabe für den Personentracker dienen. Darauf aufbauend folgt in Kapitel 5 die Suche einer Person in der häuslichen Umgebung. Diese nutzt die zuvor vorgestellten Detektoren und das Tracking-Framework. Die Erkennung von Stürzen wird in Kapitel 6 behandelt. Dieses Kapitel kann relativ unabhängig von den vorangegangen gelesen werden. In Kapitel 7 wird der Einsatz des Gesamtsystems in realen Seniorenwohnungen beschrieben. Dabei werden die Möglichkeiten und Probleme des Systems evaluiert. Das Kapitel schließt damit den von Kapitel 2 gespannten Bogen, indem gezeigt wird, wie sich die entwickelte Systemarchitektur als Teil des Gesamtsystems im häuslichen Szenario integriert. Die gewonnenen Erkenntnisse und Ergebnisse der Arbeit werden in Kapitel 8 zusammengefasst. Abschließend erfolgt ein Ausblick auf mögliche Fortführungen und Erweiterungen der vorgestellten Algorithmen.

1.3. Publikationen

Einige Teile dieser Dissertation wurden bereits in internationalen Beiträgen publiziert:

1.3.1. Publikationen des Autors mit direktem Bezug zum Thema der Arbeit

• M. Volkhardt, C. Weinrich und H.-M. Gross (2013a). “People Tracking on a Mobile Companion Robot”. In: *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics (IEEE-SMC 2013)*. Manchester, GB: IEEE Computer Society CPS, S. 4354–4359 und

• M. Volkhardt, C. Weinrich und H.-M. Gross (2013e). "Multi-Modal People Tracking on a Mobile Companion Robot". In: *Europ. Conf. on Mobile Robots (ECMR)*:
Die Publikationen beschreiben einen multi-modalen Personentracker für mobile Roboter. Teile der beschriebenen Detektionsalgorithmen finden sich Kapitel 3. Der vorgestellte Algorithmus zum Tracking ist ein Vorläufer des Personentrackers aus Kapitel 4.

• M. Volkhardt und H.-M. Gross (2013b). "Finding People in Apartments with a Mobile Robot". In: *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*. IEEE, S. 4348–4353 und

• M. Volkhardt und H.-M. Gross (2013c). "Finding People in Home Environments with a Mobile Robot". In: *Europ. Conf. on Mobile Robots (ECMR)*:
Die Veröffentlichungen beschreiben ein Verfahren zur Suche von Personen in häuslichen Umgebungen. Das Suchverfahren wird in Abschnitt 5.3 beschrieben.

• M. Volkhardt, F. Schneemann und H.-M. Gross (2013d). "Fallen Person Detection for Mobile Robots using 3D Depth Data". In: *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*. IEEE, S. 3573–3578:
In der Publikation wird eine Methode zur Erkennung gestürzter Personen auf mobilen Robotern beschrieben. Das Verfahren ist Gegenstand von Kapitel 6.

• M. Volkhardt, St. Mueller, Ch. Schroeter und H.-M. Gross (2011b). "Playing Hide and Seek with a Mobile Companion Robot". In: *IEEE-RAS Int. Conf. on Humanoid Robots (HUMANOIDS)*. IEEE, S. 40–46:
Die Publikation beschreibt ein visuelles Verfahren zur Suche von Personen an bekannten Aufenthaltsorten in der Wohnung. Als Erweiterung zu Volkhardt u. a. (2011a) unterstützen Smart-Home Sensoren die Suche. Die Methode wird in Abschnitt 5.2 beschrieben.

• M. Volkhardt, St. Mueller, Ch. Schroeter und H.-M. Gross (2011a). "Detection of Lounging People with a Mobile Robot Companion". In: *Int. Conf. on Intelligent Robotics and Applications (ICIRA)*. Bd. 7102. LNCS 2. Springer, S. 328–337:
Die Veröffentlichung beschreibt ein farb- und kantenbasiertes Verfahren zur Suche und Erkennung von Personen an bekannten Aufenthaltsorten in einer häuslichen Umgebung. Die Methode wird kurz in Abschnitt 5.2 zusammengefasst.

• M. Volkhardt, C. Weinrich, C. Schröter und H.-M. Gross (2009b). "A Con-

cept for Detection and Tracking of People in Smart Home Environments with a Mobile Robot". In: *2nd CompanionAble Workshop co-located with the 3rd European Conference on Ambient Intelligence November 18th - 21st*. Salzburg, Austria:
Die Arbeit stellt ein Konzept für das Detektieren und Tracken von Personen in häuslichen Umgebungen vor. Die vorgestellten Verfahren bilden einen Teil von Kapitel 3.

1.3.2. Publikationen als Co-Autor mit direktem Bezug zur Arbeit

• H.-M. Gross, St. Mueller, Ch. Schroeter, M. Volkhardt, A. Scheidig u. a. (2015). "Robot Companion for Domestic Health Assistance: Implementation, Test and Case Study under Everyday Conditions in Private Apartments". In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, S. 5992–5999:
Die Publikation beschreibt die Entwicklung und Evaluation des Assistenzroboters aus dem SERROGA Projekt. Dabei werden die mehrtägigen Funktions- und Nutzertests in Seniorenwohnungen beschrieben, welche teilweise Gegenstand von Abschnitt 4.10.5, Abschnitt 5.4.3 und Kapitel 7 sind.

• A. Scheidig, K. Debes, St. Mueller, Ch. Schroeter, M. Volkhardt u. a. (2015). "SERROGA: Funktions- und Nutzertests Herangehensweise und Ergebnisse". In: *German AAL Conference (AAL)*. VDE, S. 34–43:
Die Veröffentlichung beschreibt die Herangehensweise der Nutzertests aus dem SERROGA Projekt und präsentiert Ergebnisse zu Navigation, Folgeverhalten (siehe Abschnitt 4.10.5) und Personensuche (siehe Abschnitt 5.4.3).

• A. Scheidig, Ch. Schroeter, M. Volkhardt, St. Mueller, K. Debes u. a. (2014). "SERROGA: Servicerobotik fuer die Gesundheitsassistenz im nutzerzentrierten Entwurf". In: *German AAL Conference (AAL)*. VDE:
Hier werden der nutzerzentrierte Entwurf und die entwickelten Demonstratoren des SERROGA Projekts beschrieben und mit anderen Projekten verglichen (siehe Abschnitt 2.2.2).

• Ch. Schroeter, St. Mueller, M. Volkhardt, E. Einhorn, H.-M. Gross u. a. (2014). "CompanionAble – Ein robotischer Assistent und Begleiter fuer Menschen mit leichter kognitiver Beeintraechtigung". In: *German AAL Conference (AAL)*:

Diese Veröffentlichung präsentiert die Ergebnisse von Nutzertests mit dem Assistenzroboter aus dem CompanionAble Projekt. Die präsentierten Funktionen und Evaluationen bilden den Wegbereiter für das SERROGA Projekt (Gross u. a. 2015).

• C. Weinrich, T. Wengefeld, M. Volkhardt, A. Scheidig und H.-M. Gross (2014b). "Generic Distance-Invariant Features for Detection of People with Walking Aid in 2D Range Data". In: *Proc. 13th Int. Conf. on Intelligent Autonomous Systems (IAS 2014)*. Padua, Italy, S. 12:
Die Veröffentlichung beschreibt generische Merkmale zur Detektion von Personen in Laserdaten. Die Merkmale können als zusätzliche Erweiterung zu Abschnitt 3.2.2 gesehen werden.

• C. Weinrich, M. Volkhardt und H.-M. Gross (2013b). "Appearance-Based 3D Upper-Body Pose Estimation and Person Re-Identification on Mobile Robots". In: *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*. IEEE, S. 4384–4390.
In der Veröffentlichung wird die Oberfläche und Kontur eines 3D-Modells auf Grafikkarten-Shadern bewertet, um die Oberkörperpose einer Person zu schätzen. Das Verfahren kann zur Wiedererkennung von Personen eingesetzt werden.

• C. Weinrich, M. Volkhardt, E. Einhorn und H.-M. Gross (2013a). "Prediction of Human Collision Avoidance Behavior by Lifelong Learning for Socially Compliant Robot Navigation". In: *IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, S. 376–381.
Hier wird ein Verfahren zur Prädiktion von Personenbewegungen im Umfeld des Roboters präsentiert. Hierzu werden die Aufenthaltswahrscheinlichkeiten der Personen online gelernt.

• C. Schröter, S. Müller, M. Volkhardt, E. Einhorn, C. Huijnen u. a. (2013). "Realization and User Evaluation of a Companion Robot for People with Mild Cognitive Impairments." In: *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA 2013)*. IEEE, S. 1145–1151: Die Publikation stellt den Assistenzroboter und Nutzertests aus dem CompanionAble Projekt vor. Das Projekt und der vorgestellte Roboter gelten als Wegbereiter für die, in dieser Arbeit verwendete, Roboterplattform aus dem SERROGA Projekt. (Abschnitt 2.2).

• H.-M. Gross, Ch. Schroeter, St. Mueller, M. Volkhardt, E. Einhorn u. a. (2012). "Further Progress towards a Home Robot Companion for People with Mild Cognitive Impairment". In: *IEEE Int. Conf. on Systems, Man, and Cy-*

bernetics (SMC). IEEE, S. 637–644: Die Veröffentlichung gibt einen Überblick über das CompanionAble Projekt und fasst verschiedene Ergebnisse, z. B. zur Navigation und Suche (Abschnitt 5.2) zusammen.

• H.-M. Gross, Ch. Schroeter, St. Mueller, M. Volkhardt, E. Einhorn u. a. (2011b). "Progress in Developing a Socially Assistive Mobile Home Robot Companion for the Elderly with Mild Cognitive Impairment". In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, S. 2430–2437 und

• H.-M. Gross, Ch. Schroeter, St. Mueller, M. Volkhardt, E. Einhorn u. a. (2011a). "I'll Keep an Eye on You: Home Robot Companion for Elderly People with Cognitive Impairment". In: *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*. IEEE, S. 2481–2488: Die Publikationen fassen verschiedene Resultate des im CompanionAble entwickelten Roboters zusammen. Dabei werden auch erste Ergebnisse des Personentrackers (Kapitel 4) und der Personensuche (Abschnitt 5.2) präsentiert.

1.3.3. Weitere Publikationen des Autors ohne direkten Bezug zur Arbeit

• R. Stricker, St. Mueller, E. Einhorn, Ch. Schroeter, M. Volkhardt u. a. (2012b). "Konrad and Suse, Two Robots Guiding Visitors in a University Building". In: *Autonomous Mobile Systems 2012 (AMS)*. Informatik aktuell. Springer, S. 49–58

• R. Stricker, St. Mueller, E. Einhorn, Ch. Schroeter, M. Volkhardt u. a. (2012a). "Interactive Mobile Robots Guiding Visitors in a University Building". In: *IEEE Int. Symp. on Robot and Human Interactive Communication (RO-MAN)*. IEEE, S. 695–700

• M. Volkhardt, St. Mueller, Ch. Schroeter und H.-M. Gross (2010). "Real-Time Activity Recognition on a Mobile Companion Robot". In: *Int. Scientific Colloquium Ilmenau (IWK)*. ISLE Verlag, S. 612–617

• M. Volkhardt, S. Kalesse, St. Mueller und H.-M. Gross (2009a). "Maximum a Posteriori Estimation of Dynamically Changing Distributions". In: *German Conf. on Artificial Intelligence (KI)*. Bd. 5803. LNAI. Springer, S. 484–491

2. Systemarchitektur im Anwendungsszenario

Dieses Kapitel beschreibt das Anwendungsszenario der Dissertation und geht auf dessen Herausforderungen ein. In Abschnitt 2.2 wird der eingesetzte mobile Roboter beschrieben und weitere Projekte vorgestellt, die sich mit der Entwicklung von Assistenzrobotern beschäftigen. In Abschnitt 2.3 werden die Systemkomponenten und die modulare Architektur dargestellt. Zum Abschluss erfolgt in Abschnitt 2.4 eine Bewertung der Architektur in Bezug auf andere wissenschaftliche Arbeiten.

2.1. Anwendungsszenario

Als Anwendungsszenario für die Dissertation dient ein mobiler Assistenzroboter für häusliche Umgebungen. In diesem Szenario soll der Roboter Senioren in ihren täglichen Aufgaben unterstützen und eine längere Autonomie der Nutzer ermöglichen. Im Gegensatz zu kontrollierten (Labor-) Umgebungen bietet das häusliche Szenario mit seiner komplexen Beschaffenheit eine Vielzahl an Herausforderungen, für die bisher noch keine ausreichend robusten Lösungen bestehen. Abbildung 2.1 liefert einen Einblick in die Einsatzumgebung und deren Randbedingungen. Die folgenden Eigenschaften charakterisieren die häusliche Umgebung als Anwendungsszenario näher:

- **Reale Problemstellung.** Es handelt sich beim Szenario um einen real existierenden Anwendungsfall. Simulationen, Modelle und kontrollierte Umgebungen vereinfachen komplexe Vorgänge der Wirklichkeit. Verschiedene Randbedingungen und Unsicherheiten können, beziehungsweise sollen, dabei nicht modelliert werden. Im tatsächlichen Einsatz können jedoch keine Vereinfachungen für das Robotersystem getroffen werden und die Algorithmen müssen mit den gegebenen Bedingungen sinnvoll umgehen.

- **Herausforderung für mobile Roboter.** Das Realweltszenario stellt den Roboter vor eine Vielzahl von Problemen. Enge, dekorierte und möblierte Räume erschweren die Navigation des Roboters und die Wahrneh-

Abbildung 2.1.: Anwendungsszenario. Die Abbildung zeigt eine typische häusliche Umgebung, in der der Roboter agieren und mit deren Randbedingungen (enge Räume, wechselnde Beleuchtungen, Verdeckungen, variable Nutzerposen) die Algorithmen umgehen müssen[a] (GiraffPlus 2012).

[a] Nutzerin Nonna Lea mit GiraffPlus. Pressemitteilung der Europäischen Kommission: `http://europa.eu/rapid/press-release_IP-14-515_de.htm`.

mung von Personen. Letzteres wird durch uneingeschränkte, wechselnde Lichtverhältnisse (Tag, Nacht, künstliche Beleuchtung, TV), Verdeckungen und menschenähnliche Strukturen zusätzlich herausfordernd. Das anspruchsvolle Ziel ist dabei die natürliche Interaktion mit dem Nutzer in seiner Alltagsumgebung.

- **Autonomie des Robotersystems.** Da sich der Assistenzroboter direkt in den Wohnbereich und den Tagesablauf des Nutzers integrieren soll, müssen verschiedene Randbedingungen eingehalten werden. So muss das System relativ klein und wendig sein und über eine gewisse Autonomie verfügen. Der Roboter muss selbstständig und frei in der Wohnung agieren und darf den Nutzer nicht behindern. Weiterhin muss das System für die eigene Energieerhaltung sorgen. Die Entwicklung von Algorithmen zur Lösung dieser Problematiken war und ist Gegenstand paralleler Arbeiten im Umfeld des Autors (Weinrich 2016; Einhorn 2018).

- **Ausrichtung am Nutzer.** Das Assistenzsystem richtet sich an den Bedürfnissen des Nutzers aus. Ein ungezwungener und natürlich agierender Nutzer, der sich frei in der Wohnung bewegt und unterschiedliche Posen einnimmt, stellt eine Herausforderung für die Navigations- und Erkennungsalgorithmen dar. Prinzipiell ist die Varianz der Ansichten der Um-

(a) Fußgängererkennung **(b)** Nutzerposen in häuslicher Umgebung

Abbildung 2.2.: Erscheinung von Personen bei Fußgängererkennung und häuslichem Szenario. (a) die Erscheinung von Fußgängern ist aufgrund der aufrechten Pose relativ ähnlich (Dollár u. a. 2012b). (b) die Erscheinung des Nutzers variiert im häuslichen Szenario stark.

gebung und des Nutzers höher als in verwandten Szenarien. Beispielsweise ist die Erscheinung von Fußgängern im Straßenverkehr relativ ähnlich (Abbildung 2.2(a)). Im häuslichen Szenario ändert sich die Erscheinung des Nutzers jedoch oftmals gravierend, je nachdem welche Pose dieser innehat und in welcher Entfernung er sich vom Roboter befindet (Abbildung 2.2(b)). Eine besonders hohe Varianz tritt bei der Erkennung gestürzter Personen auf (Kapitel 6). Für eine Interaktion mit dem Nutzer die Echtzeitfähigkeit aller Algorithmen zu gewährleisten.

- **Nutzen und Ökonomie.** Der Roboter benötigt nützliche Dienste und Funktionen sowie einen möglichst geringen Preis, um eine Anschaffung ökonomisch sinnvoll zu machen. Daher werden effiziente Algorithmen auf Basis kostengünstiger Sensoren benötigt. Dies schließt die Verwendung einiger kostspieliger Sensoren, wie Laser-Arrays (Mozos u. a. 2010), 3D-Laser (Spinello u. a. 2010b), Time-of-Flight- und Wärmebildkameras (Cielniak u. a. 2010; Ikemura u. a. 2010b) aus. Aufgrund der 24 h Verfügbarkeit eines Assistenzroboters werden den verschiedenen Diensten, wie Telepräsenz, Erinnerungen sowie kognitiver und physischer Übungen ein hoher Nutzwert zugesprochen.

2.2. Roboter in der Assistenzrobotik

Dieser Abschnitt beschreibt den eingesetzten mobilen Assistenzroboter und führt einen Vergleich zu anderen Robotern und Projekten der Assistenzrobotik

auf.

2.2.1. Roboterplattform

Als Demonstrator für die Systemarchitektur zur Nutzerwahrnehmung wird der mobile Assistenzroboter „Max“ aus den Forschungsprojekten CompanionAble und SERROGA verwendet (CompanionAble 2008; SERROGA 2012; Gross u. a. 2015). Die wichtigsten Sensoren zur Personenwahrnehmung umfassen eine 2 Mp Kamera mit einem 180° Fischaugenobjektiv, eine Kinect 3D Tiefenkamera und einen SICK S300 Laserscanner mit 270° Öffnungswinkel. Der Roboter wird durch einen PC mit Intel i7-620M quad core Prozessor und 8 GB RAM gesteuert (Gross u. a. 2012). Eine weitergehende Beschreibung der Plattform, deren Sensorik, Aktuatoren und angebotenen Dienste findet sich in Anhang A.1.

2.2.2. Projekte in der Assistenzrobotik

Neben dem in dieser Arbeit verwendeten Roboter existieren weitere Forschungs- und Entwicklungsprojekte, die sich das Ziel gesetzt haben, einen Assistenzroboter zu entwickeln (Tapus u. a. 2007). Eine gute Systematisierung findet sich in (Robinson u. a. 2014). Aufgrund der Vielzahl von Projekten kann an dieser Stelle nur ein grober Überblick gegeben werden[1].
Eines der ersten Projekte, das sich mit Assistenzrobotern für Senioren beschäftigte, war der CMU Pittsburgh NurseBot. Hier wurden bereits Erinnerungsfunktionen, Telepräsenz und soziale Interaktionen untersucht (Montemerlo u. a. 2002). Das Cogniron Projekt untersuchte die Wahrnehmungs- und Lernfähigkeiten von unterschiedlich verkörperten Robotern (Cogniron 2004).
Bei vielen aktuellen Assistenzrobotern wird ein starker Fokus auf Telekommunikation, Telepräsenz und Fernsteuerung gelegt. Bekannte Vertreter umfassen den Roboter Giraff (2010) aus dem ExCITE (2010) Projekt (Kristoffersson u. a. 2011), dessen Verbesserung „Mr Robin“ (GiraffPlus 2012), VGo (2011), Anybots' QB (2010), Gostai Jazz (2011), Texai (2012), Double (2013) und den Roboter Padbot (2014). Meist zeichnen sich diese Roboter durch eine mobile Basis mit großem Display und Kamera aus und lassen sich durch autorisierte Personen fernsteuern. Sie sollen dabei zur Stärkung der Kommunikation des Nutzers mit Verwandten oder Pflegepersonal und zur Überwachung dienen, während die Autonomie des Roboters in den Hintergrund rückt und meist

[1] Die Erläuterungen der Projekte stammen teilweise aus Gross u. a. (2012) und Schröter u. a. (2013).

gänzlich fehlt. Daher lassen sich Funktionalitäten, wie Personenwahrnehmung, -suche und Sturzerkennung, auf keinem Roboter finden.
Projekte, die das Ziel verfolgen, einen (teil-)autonomen Assistenzroboter zu entwickeln, umfassen ALIAS (2010), HealthBot (Jayawardena u. a. 2010), Robosofts Kampai (2009) im Mobiserv (2009) Projekt, FLORENCE (Brell u. a. 2010), KSERA (2010), DOMEO (2009), EmotiRob (Saint-Aime u. a. 2007) und Robo M.D. (Ven u. a. 2010). Neuere Projekte, wie Alfred (2014), Romeo 2 (2012), die Roboter Nao und Pepper (Aldebaran 2008) und RoboDynamics' Luna (2014) fokussieren sich darauf, bekannte Dienste, wie natürliche Spracheingabe, aktive soziale Inklusion, kognitive und physische Übungen, Emotionserkennung sowie Vitalparametererfassung mithilfe von tragbaren Sensoren und Robotern marktreif zu gestalten. Die angestrebten Services der Roboter ähneln dabei denen des Rahmenprojekts SERROGA, jedoch werden von keinem Roboter alle Funktionalitäten bezüglich der Methoden dieser Dissertation erfüllt.
Einen anderen Weg beschreiten Robotikprojekte, deren Fokusse auf Manipulationsaufgaben liegen und die besonders gute, aber auch teure, Hardware einsetzen. Diese Roboter umfassen Willow Garages berühmten PR2 (2010), den Haushaltsassistent Care-O-Bot 4 (2015), den Roboter Justin des DLR (Bauml u. a. 2011) sowie den Haushaltsroboter ARMAR (Graf u. a. 2009). Einerseits werden in den Projekten eindrucksvolle Funktionen gezeigt, andererseits verbleiben die Szenarien bisher nur auf Demonstrator-Ebene ohne Bezug zum Endnutzer oder dem realen Einsatzfeld. Weiterhin fehlen häufig essenzielle Funktionen aus anderen Bereichen, wie die autonome Navigation.
Tabelle 2.1 vergleicht die Leistungsfähigkeit einiger, der oben genannten, Projekte bezüglich der zentralen Anforderungen dieser Arbeit im Rahmen des SERROGA-Projekts. Eine ausführlichere Bewertung der Projekte findet sich in Anhang A.2. Es wird ersichtlich, dass bisher kein Projekt der Assistenzrobotik alle Kriterien und Funktionen des SERROGA Roboters erfüllt. Dies liegt sowohl am unterschiedlichen Fokus der Projekte, als auch an der Schwierigkeit der einzelnen Aufgaben. In Nutzerbefragungen wurden aber gerade die Kernpunkte der Dissertation von Senioren als sinnvoll und nützlich erachtet (Huijnen u. a. 2011).

2.3. Systemarchitektur

Die Systemarchitektur beschreibt die Gesamtapplikation und das Zusammenspiel der entwickelten Module zur Personenwahrnehmung. Das Architekturkonzept wurde dabei durch das zuvor beschriebene Szenario und dessen Anforderungen (Abschnitt 2.1) beeinflusst. Im Rahmen dieser Arbeit soll die Ar-

Tabelle 2.1.: Vergleich von Projekten der Assistenzrobotik nach den Kernpunkten der Dissertation. Ein „x“ bedeutet, dass das Projekt diese Aufgabe adressiert, während ein „-“ ausdrückt, dass dieser Punkt nicht Bestandteil des Projekts ist.

Projekt	**Autonomie**	**Pers. Wahrnehmung**	**Nutzer Folgen**	**Suche**	**Sturzerkennung**	**Einsatzumgebung**
SERROGA	Autonom	x	x	x	x	Wohnung
ExCITE	Ferngesteuert	-	-	-	-	Wohnung
VGo	Teilautonom	-	-	-	-	Gemeinschaft
Mobiserv	Autonom	x	-	-	(x)	Wohnung
PR2	Autonom	x	x	-	-	Labor
Pepper	Autonom	x	-	(x)	-	Wohnung

chitektur in Anlehnung an Jaakkola u. a. (2011) und den ISO/IEC/IEEE 42010 Standard wie folgt definiert werden:

> **Definition 2.1: Systemarchitektur**
>
> Eine Systemarchitektur beschreibt den Aufbau, die Eigenschaften und die Zusammenhänge eines Systems und dessen Komponenten in der Umwelt. Sie definiert die Struktur und das Zusammenwirken der einzelnen Module und lässt Schlussfolgerungen über die Funktionsweise und den Datenaustausch zu.

2.3.1. Modulare Systemarchitektur

In Abbildung 2.3 ist die entwickelte modulare Systemarchitektur grafisch veranschaulicht. Zur Erhöhung der Übersichtlichkeit stellt die Abbildung nur die wichtigsten Komponenten und Verbindungen dar. In Wirklichkeit existieren viele weitere Elemente und Datenübertragungen, welche zum Teil in den nachfolgenden Kapiteln erläutert werden. Jede dargestellte Komponente produziert und/oder konsumiert Daten, welche innerhalb der Komponente verarbeitet werden. Die farblichen Markierungen fungieren als logische Gruppierung. Als Eingang dienen der Architektur Daten von unterschiedlichen Sensoren (rot markierte Boxen), die auf dem Roboter installiert sind (Abschnitt 2.2.1). Die Daten eines Sensors werden von einem oder mehreren Detektionsmodulen zu Personenhypothesen verarbeitet (grün markierte Boxen). Als Besonderheit

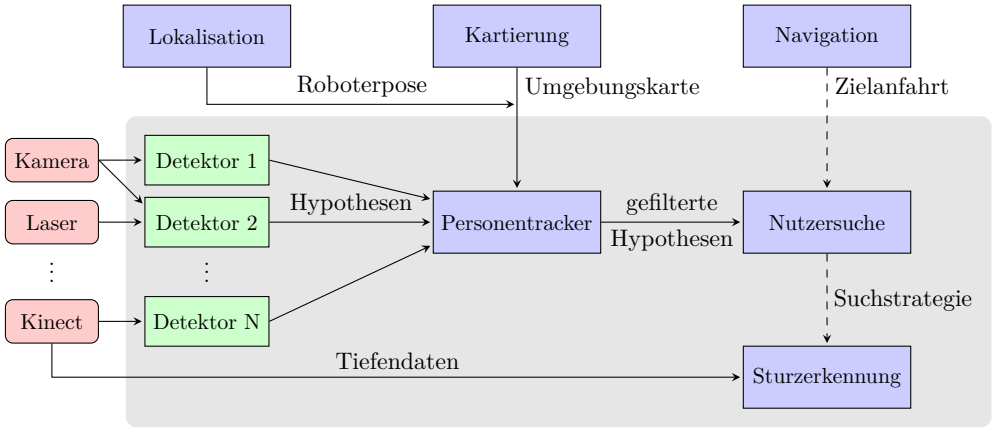
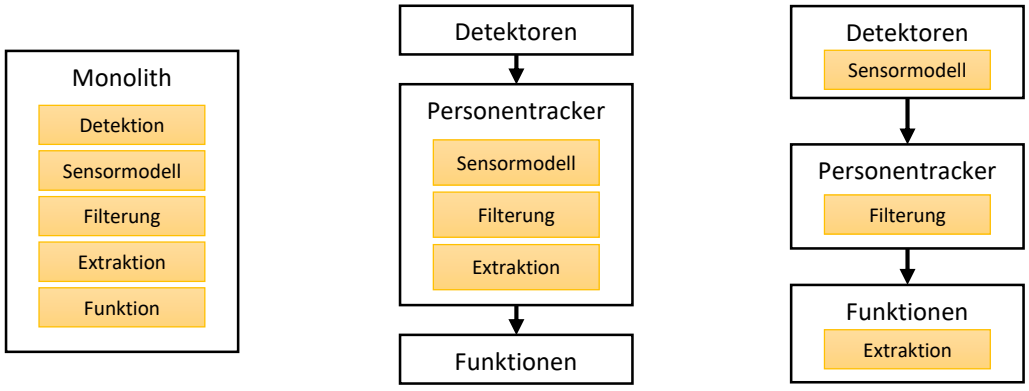


Abbildung 2.3.: Systemarchitektur. Die Grafik zeigt die in dieser Arbeit entwickelten Module (grau hinterlegt) sowie den Hauptdatenaustausch. Gestrichelte Linien bedeuten, dass keine Daten ausgetauscht werden, sondern Funktionalität genutzt wird. Module, die zur Detektion von Personen eingesetzt werden, sind grün hinterlegt. Zur Einordnung sind weitere externe Module aufgeführt (Blau; Sensoren in Rot).

wandelt jeder Detektor seine Personendetektionen mittels einer spezifischen Transformation in 3D-Hypothesen um (Abschnitt 3.5). Dadurch kommunizieren die Detektoren über eine einheitliche, asynchrone Schnittstelle mit dem Personentracker und sind austauschbar und erweiterbar (Kapitel 3). Der Tracker behandelt demnach prinzipiell alle Beobachtungen der Module gleich und erwartet nicht das Vorhandensein eines bestimmten Detektors. Neue Module lassen leicht integrieren, solange diese die gleiche Schnittstelle implementieren. Der Personentracker stellt die gefilterten Hypothesen für andere Module bereit. Die dazu verwendeten Filteralgorithmen und Systemmodelle sind austausch- und erweiterbar (Kapitel 4). Zusätzlich werden die Roboterpose und eine Umgebungskarte genutzt, die von externen Modulen zur Verfügung gestellt werden. Die resultierenden Hypothesen werden beispielsweise durch das Modul zur Nutzersuche verwendet, um festzustellen, ob der Nutzer erreicht wurde. Die Roboternavigation während der Suchfahrt durch die Wohnung übernimmt dabei ein externes Navigationsmodul. Das Modul zur Sturzerkennung nutzt Tiefendaten, um am Boden liegende Personen zu erkennen. Um die gesamte Wohnung zu erfassen, kann es sich der Algorithmen der Nutzersuche bedienen. In der Grafik sind die Teile, die in dieser Dissertation entwickelt wurden, grau hinterlegt.



(a) monolithisch **(b)** klassisch **(c)** vorgeschlagen

Abbildung 2.4.: Trackingarchitekturen. (a) Monolithische Architekturen verinnerlichen alle Funktionalität in einem Modul. (b) Klassische modulare Architekturen erlauben den Austausch von Sensoren und Funktionen, besitzen aber eine starke Abhängigkeit der Module. (c) Die vorgeschlagene modulare Architektur verlagert das Sensormodell in die Detektoren und die Extraktion von Wissen in die Funktionsmodule, um Unabhängigkeit zu ermöglichen.

2.3.2. Vergleich mit klassischen Trackingarchitekturen

In der Literatur existieren verschiedene Architekturen für das Tracken von Personen und Objekten. Die vorhandenen Methoden lassen sich in monolithische und modulare Architekturen einteilen (Abbildung 2.4). Zur Festlegung einer passenden Architektur sind neben den szenariobedingten Anforderungen auch die gewünschten Qualitätskriterien maßgeblich entscheidend. Im Rahmen der ISO/IEC 9126 werden die Qualitätskriterien einer Software-Architektur folgendermaßen angegeben:

1. Funktionalität
2. Zuverlässigkeit
3. Benutzbarkeit
4. Effizienz
5. Wartbarkeit/Änderbarkeit
6. Übertragbarkeit

Zentralisierte (monolithische) Architekturen (Abbildung 2.4(a)) zeichnen sich durch hohe *Effizienz* und einfache *Benutzbarkeit* aus und lösen meist ein spezifisches, unveränderliches Problem (Bellotto u. a. 2009; Bajracharya u. a. 2009;

Spinello u. a. 2012). Dabei sind die Anzahl und die Art der Detektoren sowie die Funktionalität der Architektur bekannt und ändern sich nicht. Das System ist hierdurch einfach implementierbar und die einzelnen Detektoren gut aufeinander abstimmbar. Der fest verbundene Aufbau der Komponenten besitzt allerdings die Nachteile einer geringen *Funktionalität* und schlechter *Wartbarkeit* und *Übertragbarkeit*.
Klassische modulare Architekturen (Abbildung 2.4(b)) verbessern die Erfüllung genau dieser Kriterien, indem die Detektoren und die Funktionalität der Anwendung aus dem Personentracker herausgelöst werden (Dietmayer u. a. 2005; Schubert u. a. 2010; Choi u. a. 2013). Allerdings ist der Personentracker immer noch stark mit den jeweiligen Detektoren verzahnt, da das spezifische Sensormodell im Personentracker liegt und jeder Detektor unterschiedliche Informationen, z. B. Bounding-Boxen, Laser- oder Radarmessungen, liefert, für die der Tracker eine Schnittstelle bereitstellen muss. Ähnlich verhält es sich mit der aufbauenden Funktionalität. Meist wird für jede zu realisierende Funktionalität das Wissen aus den Hypothesen im Tracker extrahiert und modifiziert. Daraus ergeben sich sehr anwendungsspezifische Schnittstellen.
Die vorgeschlagene modulare Architektur (Abbildung 2.4(c)) versucht diesen Punkt zu verbessern. Im Gegensatz zu klassischen, modularen Architekturen verlagert die Architektur das Sensormodell aus dem Personentracker in die Detektoren. Diese kommunizieren über eine einheitliche Schnittstelle (Hypothesen) mit dem Personentracker (Abschnitt 3.5). Weiterhin wird die Extraktion von Wissen aus den Hypothesen in die Funktionsmodule verlagert. Dadurch können sämtliche Komponenten weitgehend eigenständig arbeiten und durch den Austausch von allgemeingültigen Daten miteinander kommunizieren. Hierdurch wird der Personentracker von den Detektoren und dem Einsatzszenario beziehungsweise der Funktionalität unabhängig. Durch den hohen Grad an Modularität erfüllt die Architektur die Kriterien *Benutzbarkeit* und *Wartbarkeit*, da einzelne Module leichter parametrisiert, entfernt, hinzugefügt, ausgetauscht, gewartet oder verändert werden können. Zusätzlich lässt sich die modularisierte Architektur leicht auf andere Szenarien und Roboter *übertragen*. Dadurch erhöht sich auch die *Funktionalität* der Architektur. Je nach Anwendungsfokus kann durch das „Plug-in and out“ Konzept der Detektoren ein Kompromiss zwischen Qualität und Performance gebildet werden. Allerdings beschränkt die standardisierte Schnittstelle die Informationen, die die Detektoren an den Tracker geben können, was die *Übertragbarkeit* gegenüber klassischen Architekturen mindert. Als Alternative können Abstriche in *Wartbarkeit* und *Änderbarkeit* gemacht werden, indem die Schnittstellen angepasst werden. Das *Effizienzkriterium* wird insoweit erfüllt, dass alle Algorithmen in Echtzeit auf der gegebenen Hardware des Roboters ausgeführt werden. Einen ähnlichen modularen Ansatz verfolgt die *Fences*-Architektur von Kubertschak

u. a. (2014) für den Aufbau von statischen Karten mit unterschiedlichen Sensoren.
Für das betrachtete Szenario lassen sich die Anforderungen an den Informationsaustausch relativ gut vorab schätzen und die Vorteile der einfachen Austauschbarkeit von Funktionalität und Detektoren überwiegen. Zusätzlich erlaubt die bewusste Entscheidung zur modularen Struktur aus Abbildung 2.4(c) im behandelten Szenario eine geringere Entwicklungszeit und ermöglicht bessere Anpassbarkeit und Erweiterbarkeit. Dies ist ein klarer Vorteil gegenüber einer zentralisierten Architektur beziehungsweise klassischen modularen Architektur, da die geeignetsten Algorithmen und Parametereinstellungen vor experimentellen Funktionstests beziehungsweise der Benutzung durch die Senioren nicht bestimmt werden können. Weiterhin lassen sich so relativ einfach neue State-of-the-Art Algorithmen integrieren, falls diese eine verbesserte Detektionsqualität gegenüber den bisher verwendeten Algorithmen liefern (Abschnitte 3.3.4 bis 3.3.6). Zuletzt gewährleistet die Systemarchitektur Nachhaltigkeit im Sinne einer langlebigen, projektübergreifenden Softwarelösung zur Nutzerwahrnehmung.

2.4. Diskussion und Fazit

Dieses Kapitel stellte die entwickelte, modulare Architektur im Anwendungsszenario vor. Neben den Herausforderungen des Anwendungsszenarios und den sich daraus ergebenen Anforderungen an die Systemarchitektur wurde ein Vergleich mit anderen bestehenden Assistenzroboter-Projekten dargelegt. Die zugrunde liegenden Ideen einer modularen Systemarchitektur mit gemeinsamer Schnittstelle sind dabei keine Erfindung dieser Dissertation (Dietmayer u. a. 2005). Als Neuentwicklung wird in der vorgestellten Architektur das Sensormodell aus dem Personentracker in die Detektoren ausgelagert. Ähnlich verhält es sich mit Funktionalitäten, die in die nachfolgenden Module verschoben werden. Der Personentracker besitzt damit eine einheitliche Schnittstelle für In- und Output in Form von 3D-Hypothesen (Abschnitt 3.5).
Die meisten Architekturen von Assistenzrobotern verwenden monolithische oder klassische modulare Architekturen. Im Gegensatz dazu arbeitet der in dieser Arbeit entwickelte Personentracker jedoch mit asynchronen Beobachtungen heterogener Detektionsmodule (Kapitel 3). Da jeder Detektor als eigenständiges Modul und unabhängig vom Personentracker arbeitet, können verschiedene Verfahren schnell prototypisch implementiert, integriert und getestet werden. Somit lassen sich auch zukünftige Detektoren, beispielsweise Deep Learning Verfahren (Abschnitt 3.3.6), einfach integrieren.
Als Vorwegnahme für die nachfolgenden Kapitel erreicht das Personentracking

im schwierigen häuslichen Anwendungsszenario mit Verdeckungen und variablen Posen keine 100%tig zufriedenstellende Ergebnisse. Dennoch existiert kein vergleichbares realwelttaugliches Gesamtkonzept in der Assistenzrobotik (Abschnitt 2.2.2), das die geforderte Funktionalität des Szenarios erfüllt und mit allen Schwierigkeiten, wie Echtzeitfähigkeit, Robustheit, Handhabung von Falsch-Detektionen und Trackingfehlern, umgehen kann. Durch eine Kombination mit Dialog, Lokalisierung und Navigation bedient die Architektur ein reales Anwendungsszenario und ermöglicht ausgereifte Dienste, wie Nutzersuche, Sturzerkennung und Folgeverhalten (Kapitel 7).

3. Personendetektion

In diesem Kapitel werden die bekannten Verfahren zur Personendetektion systematisiert und anschließend Ansätze in Laser-, Bild- und Tiefendaten (Abschnitte 3.2 bis 3.4) vorgestellt. Dabei werden Vertreter, welche im Anwendungsszenario Verwendung finden, genauer erläutert und gegebenenfalls entwickelte Erweiterungen beschrieben.
Die gemeinsame Schnittstelle der Detektoren und die Generierung von 3D-Hypothesen werden in Abschnitt 3.5 erläutert. In diesem Abschnitt werden auch die Vorbereitung der Hypothesen für den Personentracker und die Auswirkung einer unsicheren Roboterpose beschrieben.

3.1. Systematisierung der Detektionsansätze

Verfahren zur Personendetektion auf mobilen Robotern gliedern sich aufgrund der verfügbaren Sensorik in drei Gruppen:

1. Verfahren auf Basis von Abstandsdaten
2. Visuelle Verfahren
3. Verfahren auf Basis von Tiefendaten.

Neben diesen Hauptgruppen existieren noch einige durchaus erfolgreiche Verfahren, die von statischer Sensorik ausgehen. Als typische Vertreter sind Hintergrundmodelle (Stauffer u. a. 1999; Schenk u. a. 2011) und Bewegungsdetektion (Schulz u. a. 2003; Müller u. a. 2007; Zhou u. a. 2012) zu nennen. Da sich mobile Roboter und deren Sensorik im Allgemeinen bewegen, werden diese Verfahren in dieser Arbeit nicht weiter betrachtet.
Geeignete Verfahren auf mobilen Robotern unterliegen außerdem den Randbedingungen der relativ geringen Hardwareressourcen sowie Echtzeitanforderungen. Im Folgenden wird näher auf die genannten Hauptgruppen eingegangen.

3.2. Detektion in Abstandsdaten

Abstandsbasierte Verfahren nutzen gemessene Werte von Distanzsensoren. Typische Vertreter sind Sonar- und Lasersensoren (Martin u. a. 2006; Müller u. a.

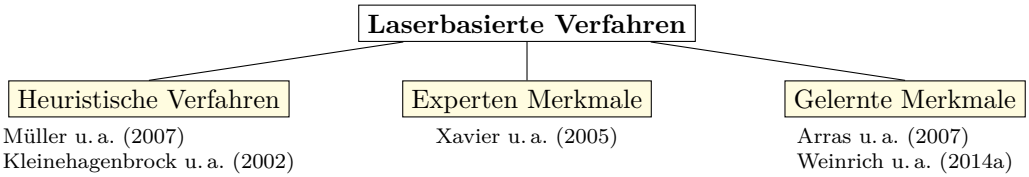


Abbildung 3.1.: Systematisierung laserbasierter Verfahren. Neben einfachen Heuristiken existieren von Experten generierte sowie datengetriebene Merkmale.

2007), wobei der Großteil der Projekte Lasersensoren nutzt. Die Vorteile von laserbasierter Personendetektion umfassen den großen Sichtbereich des Sensors sowie die geringen Unsicherheiten in der Lokalisierung der Person. Weiterhin lassen sich die Daten eines Sensors mit relativ wenig Rechenleistung verarbeiten. Der Aufbau eines Laserscans ist in Anhang A.3.1 beschrieben.

3.2.1. Systematisierung laserbasierter Ansätze

Die meisten Laserscanner befinden sich auf Höhe der Beine. Daher beschäftigen sich viele Ansätze mit der Erkennung von Beinen und Beinpaaren. Eher selten ist ein einzelner Lasersensor auf Höhe der Brust angebracht (Luo u. a. 2007). Andererseits existieren einige Arbeiten, welche Arrays von Lasersensoren einsetzen, bei denen jeweils ein Klassifikator pro Layer arbeitet (Carballo u. a. 2008; Mozos u. a. 2010). Neuere Verfahren nutzen 3D Laserscanner (Navarro-Serment u. a. 2010; Spinello u. a. 2011a). Ferner existieren noch Methoden, die Gruppen von Personen detektieren (Lau u. a. 2009) und auf einem statischen Hintergrundmodell (Shao u. a. 2008; Schenk u. a. 2011) oder auf der Detektion von Bewegungen (Schulz u. a. 2003; Mucientes u. a. 2006; Zhao u. a. 2007) basieren.
Verfahren auf mobilen Robotern mit einem Laser in Beinhöhe lassen sich in drei Gruppen einteilen (Abbildung 3.1). Nahezu alle Verfahren der drei Gruppen segmentieren den Laserscan zunächst, indem angrenzende Strahlen mit ähnlichem Abstandswert zusammengefasst werden. Die Segmenträndern können dabei mittels fester Sprungdistanz (Arras u. a. 2007), adaptiver Distanz (Premebida u. a. 2005) oder Kalman-gefilterter (Borges u. a. 2004) Distanzen berechnet werden (Abbildung A.2). Anschließend werden die einzelnen Segmente in Bein oder Hintergrund klassifiziert. Abbildung 3.1 gibt eine Übersicht über die wichtigsten Vertreter. Heuristische Verfahren bestimmen einfache Merkmale, beispielsweise die Breite eines Segments (Kleinehagenbrock u. a. 2002; Müller u. a. 2007). Verbesserte Verfahren nutzen eine Kombination aus geometrischen Merkmalen, wie die Breite, Kreisähnlichkeit und Standardabweichung

(Xavier u. a. 2005). Datengetriebene Verfahren bestimmen die besten Merkmale mittels Machine Learning Techniken (Arras u. a. 2007; Weinrich u. a. 2014a). Das Verfahren von Arras u. a. (2007) hat sich als Standard etabliert und findet in vielen Arbeiten Anwendung (Zivkovic u. a. 2007; Premebida u. a. 2009; Kondaxakis u. a. 2009; Spinello u. a. 2010b). Im nächsten Abschnitt wird eine im Rahmen der Arbeit entwickelte Erweiterung für dieses Verfahren vorgestellt.

3.2.2. Erweiterung von Arras u. a. (2007)

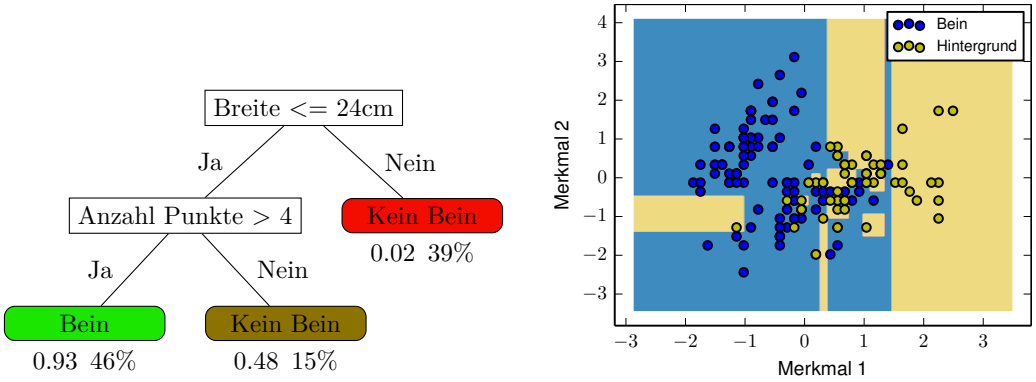
In Arras u. a. (2007) werden 14 geometrische und statistische Merkmale für die Klassifikation von Beinstrukturen in Laserdaten vorgestellt. Zur Segmentierung werden die aufeinanderfolgenden Lasermesswerte $P_i = (r_i, \phi_i)$ jeweils einem Segment zugeordnet, solange ihr gemessener Abstandswert r_i unter einer Schwelle Θ zum vorangegangen Wert liegt (Abbildung A.2). Unter den anschließend extrahierten Merkmalen befinden sich beispielsweise Breite, Linearität, Kreisähnlichkeit, mittlere Krümmung und Radius. Zur Merkmalsauswahl und zur Schwellwertbestimmung für die einzelnen Merkmale wird ein AdaBoost Algorithmus verwendet.

AdaBoost Algorithmus

AdaBoost kombiniert mehrere einfache, möglichst unabhängige, Klassifikatoren zu einem besseren Gesamtklassifikator, indem die Antworten der einzelnen Klassifikatoren statistisch gemittelt werden (Freund u. a. 1997).
Arras u. a. (2007) verwenden den klassischen AdaBoost Algorithmus (Viola u. a. 2002), bei dem sukzessive unterschiedliche Klassifikatoren trainiert werden, indem die Gewichte des Trainingsdatensatzes manipuliert werden. Die Klassifikatoren arbeiten jeweils auf genau einem Merkmal. Dies erzeugt relativ unabhängige Klassifikatoren und nimmt eine implizite Merkmalsauswahl vor. Auf der anderen Seite ist die Klassifikationsgüte der einzelnen eindimensionalen Entscheidungsfunktionen stark eingeschränkt.
AdaBoost selbst erlaubt jeden möglichen Algorithmus, der mit gewichteten Trainingsbeispielen umgehen kann. Der ausführliche Algorithmus sowie Pseudocode finden sich Anhang A.3.2.

Nutzung von Entscheidungsbäumen

Diese Arbeit schlägt als Erweiterung von Arras u. a. (2007) binäre Entscheidungsbäume als Klassifikatoren vor, welche mehr als ein Merkmal nutzen können (Breiman u. a. 1984). Abbildung 3.2 zeigt einen Entscheidungsbaum und



(a) 2-stufiger Entscheidungsbaum **(b)** Entscheidungsoberfläche

Abbildung 3.2.: Binärer Entscheidungsbaum. (a) 2-stufiger Baum für die Beinklassifikation. Entscheidungsknoten in Weiß. Blätter nach Klassenzugehörigkeit gefärbt. (b) Entscheidungsoberfläche eines Entscheidungsbaums auf 2 Merkmalen. Die Zugehörigkeit von Regionen ist in der Klassenfarbe gekennzeichnet. Grafiken erstellt mit Pedregosa u. a. (2011).

eine beispielhafte Entscheidungsoberfläche (eng. *decision surface*). In Abbildung 3.2(a) wird deutlich, dass jeder Knoten, die Daten in 2 Teilmengen einteilt, welche anschließend weiter geteilt werden können oder in einem Blatt mit einer bestimmten Wahrscheinlichkeit einer Klasse zugeordnet werden. Die Zahlen unter den Blättern geben die Sicherheit der Klassenzugehörigkeit und den prozentualen Anteil der Trainingsbeispiele, die in dieses Blatt fallen, an. Die Sicherheit der Klassifikation wird im AdaBoost Algorithmus mit dem Wichtungsfaktor α multipliziert, um eine Entscheidung zu treffen. Details zum Training und der mathematischen Formulierung eines Entscheidungsbaums sind in Anhang A.3.3 zu finden.

Generell werden für das Training eines Binärbaumes relativ viele Beispiele benötigt, da sich die Anzahl der Daten, die einem Knoten zur Bestimmung der Merkmale und Schwellwerte zur Verfügung stehen, nach jeder Partitionierung vermindert. Dies wurde im Rahmen der Arbeit ermöglicht, indem eine automatische Hintergrundsegmentierung für die Aufnahme von Trainingsdaten implementiert wurde, die das manuelle Labeln von Hand ersetzt. Nach der Klassifikation werden Beine zu Beinpaaren zusammengefasst, falls die euklidische Distanz zweier positiv klassifizierter Segmente eine empirisch ermittelte Distanz unterschreitet (z. B. die maximale Schrittweite einer Person).

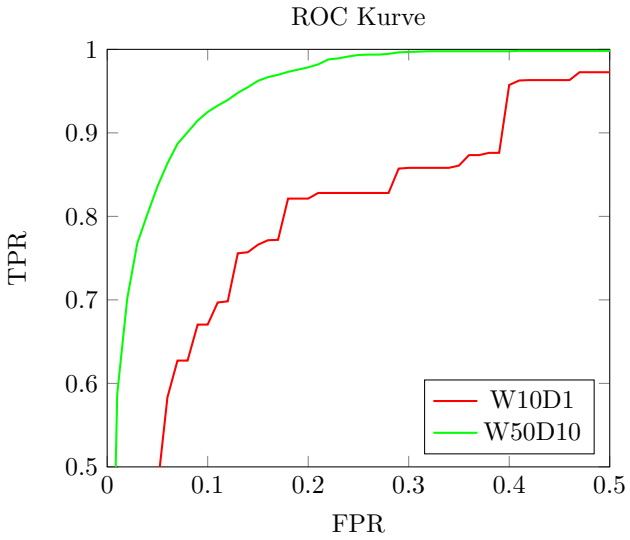
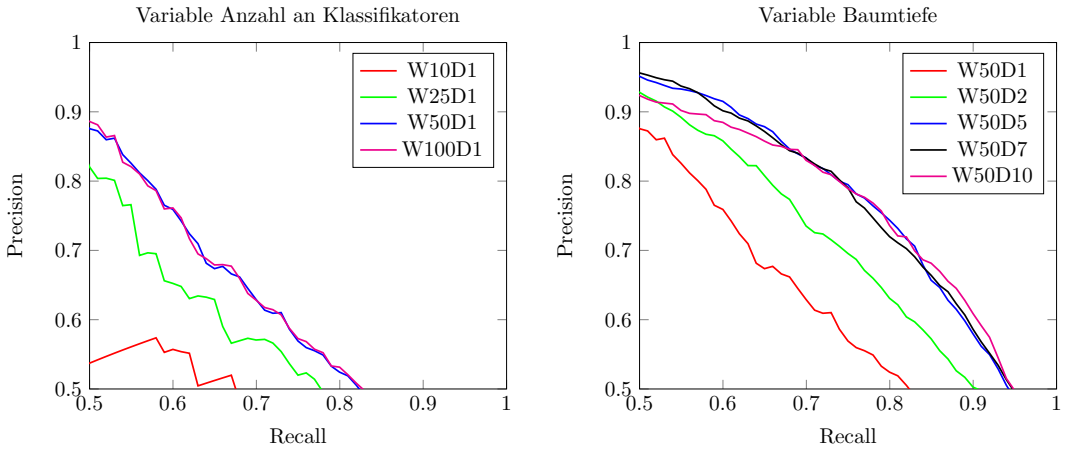


Abbildung 3.3.: ROC-Kurven des Algorithmus von Arras u. a. (2007) (Rot) und der Erweiterung um Entscheidungsbäume mit zusätzlichen Klassifikatoren (Grün). W10D1 entspricht 10 Klassifikatoren mit Tiefe 1. W50D10 entspricht 50 Klassifikatoren mit Tiefe 10. Die Erweiterung erreicht ein TPR von 92% bei einer FPR von 10% (vgl. Wengefeld (2014)).

Ergebnisse

Durch den Einsatz von binären Entscheidungsbäumen konnte die Klassifikationsgüte gegenüber Arras u. a. (2007) stark verbessert werden. Der untersuchte balancierte Testdatensatz umfasste jeweils 1250 positive und negative Beispiele. In der Arbeit wurde eine unterschiedliche Anzahl an schwachen Klassifikatoren und eine unterschiedliche Tiefe der Entscheidungsbäume untersucht. Die Ergebnisse werden mittels Receiver Operating Characteristic (ROC) bewertet (Anhang A.9.1). Die ROC-Kurve in Abbildung 3.3 zeigt die vollständige Verbesserung der Erweiterung gegenüber Arras u. a. (2007). Dieser verwendet 10 eindimensionale Entscheidungsfunktionen (entspricht einer Baumtiefe von 1). In dieser Arbeit werden 50 Binärbäume mit einer Tiefe von 10 verwendet. Die Erweiterung erreicht auf dem Testdatensatz bei einer FPR von 10% ein TPR von 92%, während das Verfahren von Arras u. a. (2007) eine TPR von 65% erreicht.
Zur tieferen Analyse visualisiert Abbildung 3.4(a) den Einfluss einer unterschiedlichen Anzahl an Klassifikatoren und Abbildung 3.4(b) stellt den Einfluss der Baumtiefe dar. In Abbildung 3.4(a) wird deutlich, dass die Klassifikationsgüte von Arras u. a. (2007) (rote Kurve) verbessert werden kann, in dem mehr schwache Klassifikatoren eingesetzt werden. In Abbildung 3.4(b)



(a) Variable Klassifikatoren **(b)** Variable Baumtiefe

Abbildung 3.4.: Einfluss unterschiedlicher Anzahl an Klassifikatoren und Baumtiefe. (a) Eine Erhöhung der Klassifikatoren erhöht die Klassifikationsgüte (W10D1 entspricht dem Verfahren von Arras u. a. (2007)). (b) Eine Erhöhung der Baumtiefe verbessert die Klassifikationsgüte beachtlich (vgl. Wengefeld (2014)).

zeigt sich, dass das Ergebnis durch leistungsfähigere schwache Klassifikatoren (Binärbäume) mit mehrfachen Entscheidungen (Baumtiefe) stark verbessert werden kann. Eine Baumtiefe von fünf liefert den besten Kompromiss aus Performance und Klassifikationsgüte.
Die Leistung des Verfahrens in der Trackingarchitektur im Anwendungsszenario wird in Abschnitt 4.10.3 evaluiert. Für eine ausführlichere Evaluation sowie einer Verbesserung der verwendeten Merkmale sei auf Wengefeld (2014)[1] und Weinrich u. a. (2014b) verwiesen.

3.3. Detektion in Bilddaten

Visuelle Verfahren nutzen die Bilder von RGB-Kamerasensoren. Die Palette der eingesetzten Kameras reicht von günstigen Webcams bis hin zu Industriekameras. Häufig wird ein großer Öffnungswinkel bevorzugt oder mehrere Kameras eingesetzt, um das Sensorfeld des Roboters zu vergrößern. Bilddaten liefern im Vergleich zu anderen Sensoren sehr viele Informationen. Die Verarbeitung dieser Daten ist jedoch auch mit einem größeren rechentechnischen Aufwand verbunden. Daher lassen sich viele aktuelle visuelle Verfahren

[1] Vom Autor im Rahmen dieser Arbeit betreut.

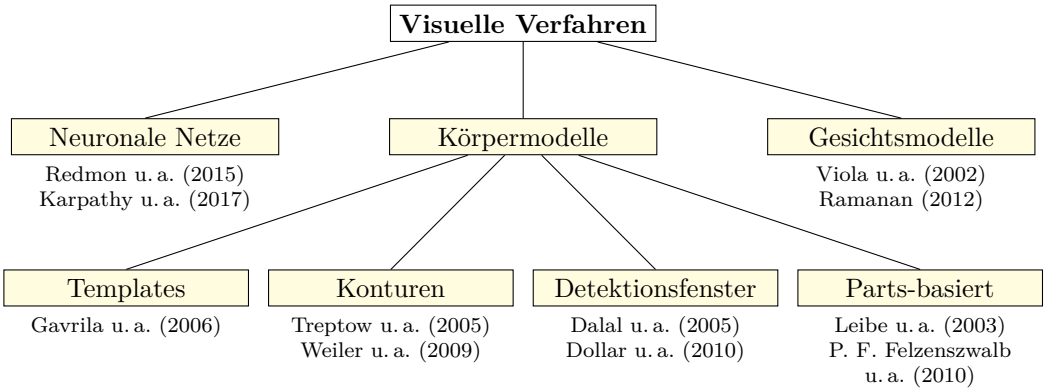


Abbildung 3.5.: Systematisierung visueller Verfahren. Erläuterungen im Text.

auf mobilen Robotern mit deren eingeschränkten on-board Rechenkapazitäten nur bedingt einsetzen.

3.3.1. Systematisierung visueller Ansätze

Das große Forschungsinteresse auf dem Gebiet der Personendetektion in Bilddaten brachte in den vergangenen Jahren eine Vielzahl von Verfahren hervor. Im Folgenden kann daher nur ein Teil der bekannten Ansätze vorgestellt werden.
Abbildung 3.5 gliedert die visuellen Verfahren in Gesichtsmodelle und Körpermodelle. Bekanntester Vertreter in der Kategorie der Gesichtsdetektion ist der Ansatz von Viola u. a. (2002). Dieser nutzt einfache Haar-Features und eine AdaBoost-Kaskade, um Gesichter effizient im Bild zu detektieren (Abschnitt 3.3.2). Neuere Verfahren nutzen parts-basierte Ansätze zur Detektion gedrehter Gesichtsposen (Ramanan 2012). Für eine Übersicht verschiedener Verfahren sei auf Zhang u. a. (2010) verwiesen.
Körpermodelle lassen sich in Templates, Konturen, Dektektionsfenster und parts-basierte Modelle einteilen. Statische Templates haben Schwierigkeiten, die hohe Varianz in der Erscheinung von Personen zu erfassen. Daher setzen Gavrila u. a. (2006) eine Vielzahl von verschieden artikulierten Silhouetten ein (Abbildung 3.6(a)). Konturbasierte Ansätze ersetzen die Steifheit von Templates durch flexible Strukturen, die dynamisch an den Körper angepasst werden (Treptow u. a. 2005; Weiler u. a. 2009). Dadurch erfordern sie jedoch eine hohe Rechenleistung und werden durch Hintergrundstrukturen beeinflusst.
Prominentester Vertreter der merkmalsbasierten Ansätze mit Detektionsfenstern sind die Histogramme orientierter Gradienten (HOG) (Dalal u. a. 2005).

(a) Templates **(b)** HOG **(c)** Parts-basiert **(d)** ISM

Abbildung 3.6.: Körpermodelle. (a) Vielzahl von Templates (Gavrila u. a. 2006), (b) Blockweise Histogramme orientierter Gradienten (Dalal u. a. 2005). (c) Körperteile als Baumstruktur (Ramanan u. a. 2005), (d) Implicit shape model mit Grauwert-Merkmalen und Votes für Objektzentren (Leibe u. a. 2003). Quelle: Volkhardt u. a. (2009b)

Die HOG (Abschnitt 3.3.3) berechnen blockweise Histogramme von Gradientenorientierungen und erreichen somit Robustheit gegenüber leichten örtlichen Variationen und Beleuchtung (Abbildung 3.6(b)). Dollar u. a. (2010) ermittelt die besten Merkmale mittels Boosting, interpoliert die Auflösungspyramide und nutzt Soft-Kaskaden, um eine beeindruckende Qualität und Geschwindigkeit zu erreichen (Abschnitt 3.3.5). Parts-basierte Ansätze extrahieren die Merkmale der Körperteile einer Person (z. B. Kopf, Torso, Gliedmaßen) und setzen diese in Relation zueinander. Die Anzahl und die Positionen der Teile können dabei fest vorgegeben wie in Abbildung 3.6(c) (Ramanan u. a. 2005) oder datengetrieben gelernt wie in Abbildung 3.6(d) (Leibe u. a. 2003; P. F. Felzenszwalb u. a. 2010) sein (siehe Abschnitt 3.3.4). Die Merkmale für die einzelnen Körperteile variieren dabei zwischen Farbhistogrammen, SIFT-Features, Kantenhistogrammen, Konturkontext und vielen Weiteren. Die parts-basierten Verfahren erreichen die besten Detektionsergebnisse bei unterschiedlichen Posen und Verdeckungen. Nachteil ist ihr relativ hoher Berechnungsaufwand, da im Allgemeinen Berechnungen für jedes Körperteil durchgeführt werden müssen (Volkhardt u. a. 2009b).

Viele aktuelle Verfahren für stehende Personen stammen aus dem Bereich der Fußgängerdetektion (Benenson u. a. 2012). Dollár u. a. (2012b) geben einen ausführlichen Vergleich von 16 Verfahren auf sechs Outdoor Datensätzen. Unter den untersuchten Methoden befinden sich auch einige der in diesem Abschnitt vorgestellten (Viola u. a. 2002; Dollar u. a. 2010; P. F. Felzenszwalb u. a. 2010). Auf demselben Datensatz, werden in Benenson u. a. (2014) über 40 Verfahren untersucht. Im Folgenden wird eine Auswahl an Verfahren vorgestellt, die im Rahmen dieser Arbeit untersucht wurden und auf dem mobilen Roboter zur Anwendung kommen.

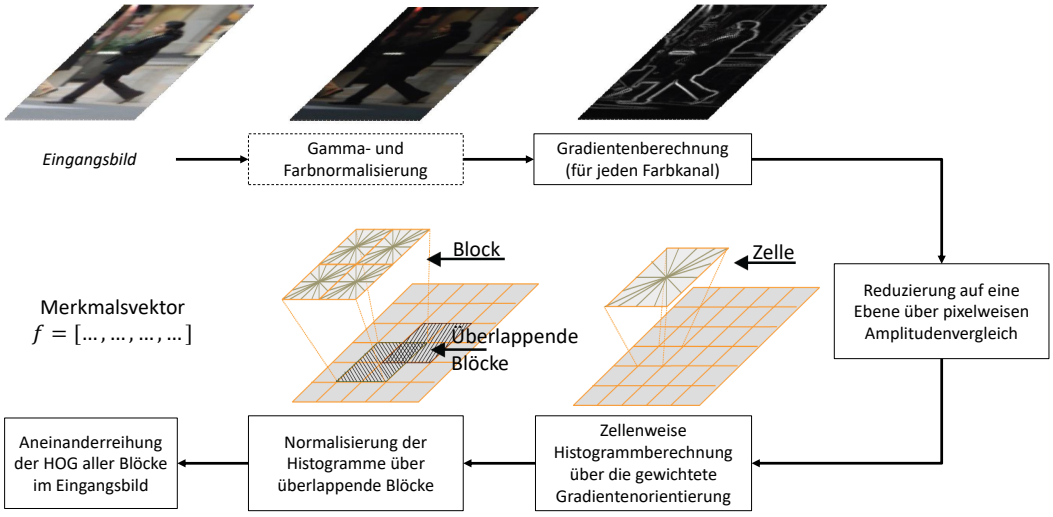


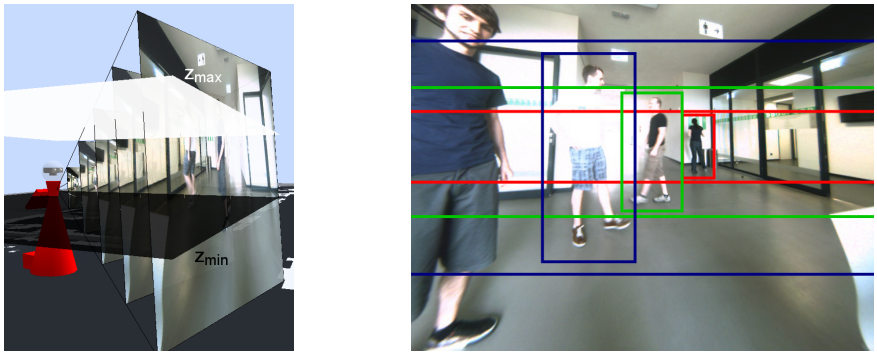
Abbildung 3.7.: Ablauf zur Berechnung der Histogramme orientierter Gradienten. Quelle: Laschka (2013)

3.3.2. Viola&Jones Gesichtsdetektor

Das Verfahren von Viola u. a. (2002) detektiert Gesichter mittels Sliding-Window Technik und Haar Merkmalen (Intensitätsfeatures). Zur Detektion wird eine effiziente kaskadierte Variante des AdaBoost Algorithmus (Abschnitt 3.2.2) verwendet. Dieser Ansatz kann Personen nur erkennen, wenn das Gesicht frontal oder leicht seitlich im Bild zu sehen ist und eine gewisse Mindestgröße besitzt. Vom Roboter abgewandte oder zu weit entfernte Personen können nicht detektiert werden. Zur Beschleunigung des Verfahrens wurde eine Region of Interest (ROI) eingeführt, die den Arbeitsbereich auf einen Bildausschnitt beschränkt. Da Gesichter von stehenden und sitzenden Personen mit der verwendeten Kamera nur in der oberen Bildhälfte auftreten, beschleunigt dies das Verfahren um Faktor zwei und verhindert Falsch-positive im unteren Bildbereich.

3.3.3. Histogramme orientierter Gradienten

Die durch die SIFT Merkmale (Lowe 2004) inspirierten Histogramme orientierter Gradienten (HOG) wurden erstmals von Dalal u. a. (2005) vorgestellt. Sie beinhalten normalisierte Histogramme der Häufigkeitsverteilung der Gradientenorientierung in einem Bildbereich. In Abbildung 3.7 ist der prinzipielle Ablauf der HOG Merkmals-Berechnung in einem Detektionsfenster dargestellt. Nach einer Vorverarbeitung und der Gradientenberechnung wird das



(a) 3D Detektionskorridor **(b)** 2D Suchbereiche

Abbildung 3.8.: Beschränkung auf Bodenebene. (a) zeigt die Auflösungspyramide in 3D mit zugehörigem Korridor. Dieser ist durch die zum Boden parallelen Ebenen z_{min} und z_{max} definiert und beinhaltet alle vertikalen Mittelpunkte von Personen. (b) stellt exemplarisch ins Originalbild projizierte Korridore für drei unterschiedliche Skalierungsstufen dar.

Bild in 8x8-Pixel große Zellen eingeteilt, in denen über den mit der Amplitude gewichteten Gradientenorientierungen ein Histogramm berechnet wird. Diese Histogramme werden anschließend durch überlappende Blöcke von jeweils vier Zellen normalisiert. Durch die Aneinanderreihung der Blöcke entsteht der HOG Merkmalsvektor (Laschka 2013). Durch das zellen- und blockweise Zusammenfassen der Orientierungen sind die HOG invariant gegenüber leichten Variationen der Körperpose und -form, Beleuchtungsschwankungen und verändertem Hintergrund. In dieser Arbeit werden ein Ganzkörper HOG Detektor (Dalal u. a. 2005) und ein Oberkörper HOG Detektor (Ferrari u. a. 2008) eingesetzt.

Beschränkung auf die Bodenebene

Der Oberkörper- und Ganzkörper HOG Detektor wurden im Rahmen dieser Arbeit um eine Beschränkung auf die Bodenebene (engl. *Groundplane Constraint*) erweitert (Volkhardt u. a. 2013a). Beim Sliding-Window Verfahren wird das Detektionsfenster an jeder Position auf jeder Skalierungsstufe der Auflösungspyramide angewendet. Dadurch lassen sich mit demselben Detektionsfenster Personen unterschiedlicher Größe beziehungsweise Entfernung erfassen. Allerdings werden auch viele Positionen untersucht, an denen sich keine Person befinden kann. Nimmt man an, dass sich Personen in einer Ebene parallel zum Boden bewegen, müssen viele Positionen, z. B. kleine Personengrößen am oberen Bildrand, nicht untersucht werden. Abbildung 3.8(a) stellt einen Robo-

ter auf einer grauen Ebene dar. Die Auflösungspyramide ist in Blickrichtung der Kamera dargestellt. Über Kenntnis der 3D Position des Roboters und der Kamera (extrinsische Parameter) kann ein Korridor festgelegt werden, in dem Personen auftreten können. Im Rahmen dieser Arbeit wurde der Bereich so gewählt, dass der Mittelpunkt der zu detektierenden Personen innerhalb eines Korridors $Z_{obj} \in [z_{min}, z_{max}]$ liegen muss. z_{min} und z_{max} beschreiben dabei zur Bodenebene parallelen Ebenen (Abbildung 3.8(a)). Der Korridor wird bewusst etwas weiter als nötig definiert, um die Kameraneigung während der Bewegung zu kompensieren. Beim Oberkörperdetektor wird der Bereich so groß gewählt, dass stehende und sitzende Personen darin erfasst werden. Mithilfe der intrinsischen Kameraparameter kann dieser Korridor auf die jeweilige Stufe der Auflösungspyramide projiziert werden. Abbildung 3.8(b) stellt den Korridor für drei unterschiedliche Skalierungsstufen im Originalbild dar. Je größer das Bild der Auflösungspyramide (entspricht einem kleineren Detektionsfenster), desto größere Teile des Bildes lassen sich auslassen. Hierdurch wird das Verfahren nahezu um den Faktor zwei beschleunigt und Falsch-positive werden reduziert. Eine allgemeine Verfahrensweise zur Bestimmung der Bodenebene für Sliding-Window Verfahren ist in Sudowe u. a. (2011) zu finden.

3.3.4. Deformierbare körperteilbasierte Modelle

Die diskriminativ trainierten deformierbaren parts-basierten Modelle (kurz DPM) wurden von P. F. Felzenszwalb u. a. (2010) vorgestellt. Sie erreichten in der PASCAL Visual Object Classes (VOC) Challenge[2] 2007 in zehn von zwanzig Kategorien die besten Detektionsergebnisse. Auch in darauf folgenden Jahren erzielte das Verfahren sehr gute Resultate. Es galt bis zum Durchbruch von Deep Learning Verfahren als Referenzverfahren für die Detektion von Personen in unterschiedlichen Posen unter partiellen Verdeckungen.
Das DPM nutzt einen Root-Filter und sogenannte Part-Filter (Abbildung 3.9). Der Root-Filter gleicht einem HOG-Deskriptor (Abschnitt 3.3.3), während die Part-Filter analog verschiedene Körperteile erfassen. Die erlaubten Positionen der Part-Filter sind flexibel, werden jedoch durch eine ideale Position (Ankerpunkte) und eine Kostenfunktion beschränkt.
Der Root-Filter beschreibt grob das Aussehen des kompletten Objekts, während die Part-Filter weitere Teile genauer kennzeichnen. Zur Detektion werden der Root-Filter und alle Part-Filter auf der HOG Auflösungspyramide getestet. Der Score einer Detektion berechnet sich aus der Passfähigkeit des Root-Filters

[2]Die VOC Challenge bietet ein einheitliches Evaluationsframework für Objektdetektoren. Unter den Objektkategorien befinden sich unter anderem Pkws, Gebrauchsgegenstände und Personen in alltäglichen Posen.

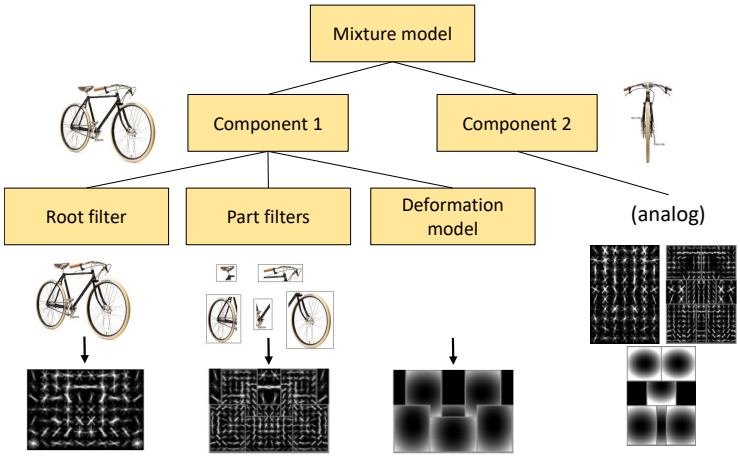


Abbildung 3.9.: Mixture Model aus deformierbaren parts-basierten Modellen. Für Objekte mit verschiedenen Erscheinungsformen werden mehrere Komponenten eingesetzt. Diese besitzen jeweils einen Root-Filter, Part-Filter und Deformationskosten für die Abweichung der Part-Filter von der idealen Position. Niedrige Kosten in Schwarz, hohe Kosten in Weiß[3]. Quelle: P. F. Felzenszwalb u. a. (2010) und Reuther (2011)

und der einzelnen Parts abzüglich der Deformationskosten. Über eine Kaskadierung der zu testenden Part-Filter kann das Verfahren bei gleicher Detektionsqualität beschleunigt werden (P. F. Felzenszwalb u. a. 2010). Zusätzlich stellen P. F. Felzenszwalb u. a. (2010) eine Dimensionsreduktion der Merkmale durch Principle Component Analysis (PCA) vor. Details zum Algorithmus finden sich in Anhang A.3.4.

Die flexiblen Part-Filter ermöglichen es, Personen in leicht unterschiedlichen Posen und Erscheinungen zu erkennen, da jede der vielen verschiedenen Part-Konfiguration quasi ein eigenständiges festes Template (engl. *rigid model*) synthetisiert. In Abbildung 2.2(b) wurde jedoch deutlich, dass sich die Erscheinungen und Posen im häuslichen Szenario stark unterscheiden. P. F. Felzenszwalb u. a. (2010) setzen daher Modelle mit mehreren Komponenten (engl. *Mixture Models*) ein. Jede Komponente entspricht dabei einem vollen Modell mit Root-Filter, Part-Filter und Deformationskosten (Abbildung 3.9). Die Komponenten erfassen beispielsweise eine aufrecht stehende Person oder nur den Oberkörper einer Person. Sie werden im Training durch Clusterung der Seitenverhältnisse der gelabelten Bounding-Boxen gewonnen.

Im Rahmen dieser Arbeit wurde zunächst die kaskadierte Variante des Algorithmus auf Basis der vorhandenen MATLAB Implementierung in C++ umgesetzt und als Detektionsmodul in die Architektur integriert. Dies erlaubte die Verarbeitung eines Bildes in VGA Auflösung in weniger als einer Sekunde

(Laschka 2013)[4]. Dadurch konnte das Verfahren on-demand auf dem mobilen Roboter eingesetzt werden, um Personenhypothesen während der Personensuche durch das DPM zu verifizieren (Abschnitt 5.3). Im weiteren Verlauf der Arbeit wird eine Erweiterung des DPM benutzt, welche die Berechnung der Merkmale im Fourier Raum durchführt (Dubout u. a. 2012). Der größte Berechnungsaufwand des ursprünglichen Verfahrens entsteht durch die Faltung der Filter mit der Auflösungspyramide. Durch die Linearität der Fourier Transformation wird diese beschleunigt und muss nicht für jedes Merkmal neu berechnet werden. Eine zusätzliche speicher- und berechnungssparende Implementierung beschleunigt das Verfahren insgesamt um Faktor 7.
Mithilfe von Einschränkung der durchsuchten Skalen der Auflösungspyramide und einer separaten CPU ist das Verfahren in Echtzeit auf dem mobilen Roboter nutzbar. Das entstandene Detektionsmodul verbessert das Personentracking erheblich (Abschnitt 4.10.3) und ermöglicht neue Suchstrategien (Abschnitt 5.4).
In parallelen Arbeiten erreichen Cho u. a. (2012) auf einer dedizierten CPU eine echtzeitfähige Version des Detektors, indem eine effiziente C Implementierung mit Bodenebene genutzt wird. Eine weitere häufig eingesetzte Erweiterung des Verfahrens ist die Einschränkung des Suchraums durch eine effiziente Ermittlung von Detektions-Kandidaten (engl. *Detection proposals*). Diese werden beispielsweise ermittelt, indem Regionen mit einer Mindestanzahl an Kanten oder Struktur gefordert werden (Hosang u. a. 2014).

3.3.5. Fastest Pedestrian Detector in the West

Dollar u. a. (2010) stellen ein Verfahren zur Detektion von Fußgängern vor, welches an vielen Stellen des Detektionsprozesses ansetzt, um die Geschwindigkeit der Personendetektion zu steigern. Zunächst werden Integral Channel Features verwendet (Dollár u. a. 2009). Diese bestehen aus verschiedenen Merkmalen, wie Haar-, LUV-Farb-, Grauwert-, Difference of Gaussians-, Kanten-, Gabor- und HOG-ähnliche Merkmale, die jeweils auf einem Integralbild berechnet werden. Zur Merkmalsauswahl und Positionierung im Detektionsfenster wird ein AdaBoost-Algorithmus mit Entscheidungsbäumen verwendet (ähnlich wie in Abschnitt 3.2.2). Die ausgewählten Merkmale werden nicht für jede Skalierung der Merkmalspyramide berechnet, sondern nur für jede Oktave[5]. Die Merkmale der Skalierungsstufen zwischen den Oktaven werden aus denen der Oktaven interpoliert.

[4] Vom Autor im Rahmen dieser Arbeit betreut.
[5] Eine Oktave entspricht einer jeweiligen Halbierung der Bildauflösung. Zwischen zwei Oktaven existieren typischerweise 10-14 weitere Skalierungsstufen.

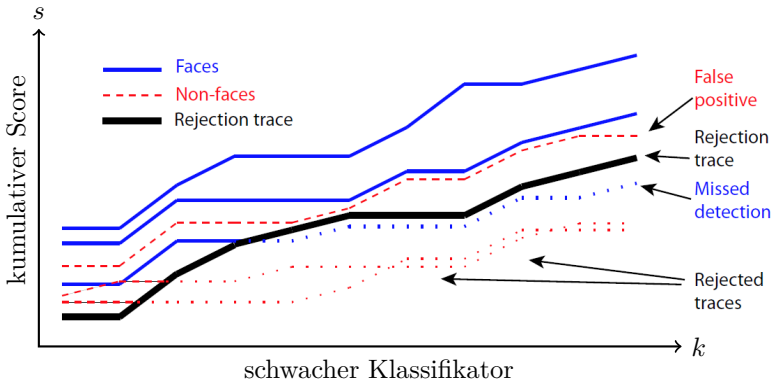


Abbildung 3.10.: Evaluation mit einer Soft-Kaskade. In Schwarz sind die einzelnen Schwellwerte t_j der schwachen Klassifikatoren dargestellt. Fällt der Score eines Beispiels nach einem Klassifikationsschritt unter diese Schwelle, wird es verworfen. Ein Beispiel muss nicht von jedem Klassifikator positiv klassifiziert werden, wenn vorherige Klassifikatoren den Gesamtscore bereits angehoben haben (waagerechte Liniensegmente). Quelle: Bourdev u. a. (2005)

Weiterhin verwenden Dollar u. a. (2010) eine sogenannte Soft-Kaskade. Der Nachteil von klassischen Kaskaden ist der Informationsverlust zwischen den einzelnen Kaskadenstufen. Das heißt, dass die Entscheidung, ob eine Hypothese verworfen oder gehalten werden soll, nicht von vorherigen Stufen abhängt. Die Soft-Kaskade nutzt nur eine Kaskadenstufe, führt jedoch für jeden der nacheinander angewendeten schwachen Klassifikatoren einen Schwellwert zum frühzeitigen Ausschließen von schlechten Hypothesen ein. Fällt der akkumulierte Score eines Beispiels während der Klassifikation durch die schwachen Klassifikatoren unter einen Schwellwert t_j, so ist für die nachfolgenden Klassifikatoren der Gesamtschwellwert T nicht mehr zu erreichen (Abbildung 3.10). Intuitiv sind die Schwellwerte die minimalen Scores, die durch die schwachen Klassifikatoren im Trainingsdatensatz für positive Beispiele erzielt wurden.
Dollár u. a. (2012a) erweiteren das Verfahren um *Crosstalk Cascades*. Durch die Verwendung von Informationen benachbarter Positionen und Skalierungen kann die Berechnung in Bereichen von schwachen Nachbarn gehemmt oder von vielversprechenden Nachbarn angeregt werden. Durch Testen eines spärlichen Sets an Positionen lassen sich viele Regionen ausschließen (z. B. in homogenen Strukturen auf Wänden).
Im Rahmen dieser Arbeit wurde die vorhandene MATLAB Implementierung nach C++ portiert und in die Architektur integriert[6]. Das Detektionsmodul dient als performante Ergänzung für die Detektion von Personen in aufrechter

[6]Umsetzung durch Alexander Katzmann.

Pose (Abschnitt 4.10.3).

3.3.6. Deep Learning Verfahren

In den vergangenen Jahren fokussierte sich das Themenfeld der Klassifikation und Detektion auf Deep Learning Verfahren. Durch die Verfügbarkeit von großen Datenmengen und leistungsfähigen GPUs konnten seit Krizhevsky u. a. (2012) durch Simonyan u. a. (2014) und Szegedy u. a. (2015) vor allem durch Deep Convolutional Neural Networks (CNN) im Bereich der visuellen Detektion beachtliche Fortschritte erzielt werden. Generative Adversarial Networks (Goodfellow u. a. 2014) nutzen ein generatives und ein entscheidendes Netzwerk, welche sich gegenseitig verbessern. In Ren u. a. (2015) schlägt ein Neuronales Netz Regionen im Bild vor, auf denen ein CNN Objekte detektiert. Karpathy u. a. (2017) kombinieren Recurrent Neural Networks (RNN) mit CNN um natürliche Beschreibungen von Bildern und deren Inhalt zu generieren.
Trotz der beeindruckenden Ergebnisse der Netze sind viele Verfahren nicht echtzeitfähig und an eine leistungsfähige GPU gebunden. Zur Zeit der Bearbeitung war auf dem Roboter keine GPU verfügbar, beziehungsweise die Verfahren noch nicht auf dem Roboter einsetzbar. Daher sind Deep Neural Network Verfahren nicht Bestandteil dieser Arbeit. Dennoch werden diese Verfahren das Themenfeld in den nächsten Jahren dominieren. Effiziente Verfahren, wie die in Redmon u. a. (2015, 2016) vorgestellten, sollten daher untersucht und integriert werden. Aufgrund der modularen Architektur Kapitel 2 lassen sich neu entwickelte Verfahren einfach als zusätzlicher Detektor nutzen.

3.4. Detektion in Tiefendaten

Tiefendaten können in Form einer 3D Punktewolke (engl. *point cloud*) oder als Tiefenbild gegeben sein. Punktewolken geben die belegten Zellen mit kartesischen Koordinaten (x, y, z) an. Bei Tiefenbildern existiert für jeden Bildpixel ein Tiefenwert, der der Entfernung des Punktes auf der Bildebene zum korrespondierenden Punkt in der Welt entspricht. Die Vorteile der Punktewolken und Tiefenbilder gegenüber RGB-Bildern umfassen die Unabhängigkeit von Farb-, Textur- und Beleuchtungsschwankungen sowie die Einschränkung von Detektionsbereichen, z. B. durch Extraktion der Bodenebene oder Nutzung der bekannten Entfernung (Munaro u. a. 2012; Munaro u. a. 2014).
Tiefenbilder können *passiv* monokular bewegungs- oder strukturinduziert entstehen. Häufiger werden binokulare oder multiokulare Kameras genutzt und die Disparität zwischen den einzelnen Bildern ermittelt. Bei den *aktiven* Sys-

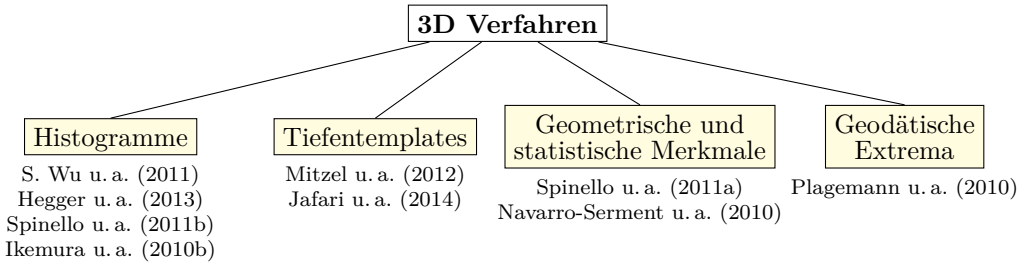


Abbildung 3.11.: Verfahren zur Personendetektion in Tiefendaten.

temen existieren Time-of-flight 3D-Laser und Kameras (Kinect 2) sowie die Methode des strukturierten Lichts (Kinect 1).

3.4.1. Systematisierung tiefenbild-basierter Verfahren

Verfahren zur Detektion in Tiefendaten sind häufig von den RGB-Verfahren inspiriert. Abbildung 3.11 gibt einen Überblick der aktuellen Verfahren auf Tiefendaten. Die Anwendung von Templates auf dem Tiefenbild wird in Mitzel u. a. (2012) und Jafari u. a. (2014) beschrieben. Die den Gradientenhistogrammen von Dalal u. a. (2005) ähnelnden Tiefenhistogramme diskretisieren Gradienten von Tiefensprüngen in einem Detektionsfenster (S. Wu u. a. 2011; Spinello u. a. 2011b). Zusätzlich existieren Histogrammfeatures, die auf der Oberfläche von Punktewolken (engl. *surface histograms*) gebildet werden (Ikemura u. a. 2010b; Hegger u. a. 2013). Die ebenfalls aus dem 2D-Bereich bekannten geometrischen und statistischen Merkmale (Arras u. a. 2007) werden in Navarro-Serment u. a. (2010) und Spinello u. a. (2011a) auf 3D Punktewolken angepasst. Weiterhin bietet die Punktewolkenoberfläche die Möglichkeit, geodätische Distanzen zur Personendetektion zu nutzen (Plagemann u. a. 2010).

3.4.2. Eignung der Verfahren für den mobilen Roboter

Im Rahmen dieser Arbeit wurden zwei Verfahren entwickelt, um Personen auf Basis von Tiefendaten zu detektieren. Suck (2013)[7] verbindet die Segmentierung von Hordern u. a. (2010) mit den Histogrammen orientierter Tiefen von Spinello u. a. (2011b). Horevych (2014)[7] implementiert und erweitert das Verfahren von Hegger u. a. (2013), welches auf Histogrammen lokaler Oberflächennormalen beruht, für die Detektion von stehenden und sitzenden Personen.
Leider erreicht keines der beiden Verfahren zufriedenstellende Ergebnisse in

[7]Vom Autor im Rahmen dieser Arbeit betreut.

(a) Stehend fern **(b)** Stehend nah **(c)** Sitzend

Abbildung 3.12.: Sichtbereich der Kinect im Vergleich zur RGB-Kamera. Die gemessene Tiefe ist in Falschfarben dargestellt. Ungültige Tiefenwerte in Weiß. Aufgrund der Einbaulage und des geringen Öffnungswinkels können Personen mit Kopf-Schulter-Templates nur sitzend aus relativ fixer Entfernung erkannt werden.

der Detektionsqualität. Dies liegt einerseits an mangelhaften Trainingsdaten für die Algorithmen (Spinello u. a. 2011b). Aufgrund einer relativ großen Entfernung der Personen zum Sensor entstehen in den Daten viele ungültige Messwerte[8] und die Tiefenauflösung in der Entfernung ist relativ gering. Andererseits ist das Sensorfeld und die Anbringung der Kinect am verwendeten Roboter (Abschnitt 2.2.1) suboptimal. Abbildung 3.12 visualisiert das Sichtfeld der Kinect im Vergleich zur genutzten RGB-Kamera. Alle in Abschnitt 3.4.1 vorgestellten Verfahren nehmen an, dass die gesamte Person oder zumindest ihr Oberkörper sichtbar und unverdeckt ist. Durch den geringen Öffnungswinkel, die relative geringe Messentfernung und Auflösung lassen sich Personen im Sichtbereich der Kinect nur in sitzender Pose und in geringer Entfernung zum Roboter erfassen[9].
Als weiteres schwer wiegendes Hindernis litt die verwendete Roboterplattform unter Hardwareproblemen, die den Kinectsensor nach kurzer Zeit ausfallen ließen. Dies führte dazu, dass ein verlässlicher Einsatz der Kinect und der 3D Verfahren auf dem Roboter nicht möglich war. Prinzipiell bieten 3D Verfahren jedoch einen nicht zu unterschätzenden Zugewinn an Informationen für die Personendetektion. Robuste Tiefenverfahren lassen sich, einen funktionierenden Sensor mit entsprechend großem Sichtbereich (z. B. Kinect 2) und rechenstarke Hardware vorausgesetzt, einfach in die bestehende Architektur integrieren.

[8] Falls sich für einen Pixel keine Korrespondenz zum Specklemuster finden lässt, liefert die Kinect einen ungültigen Tiefenwert.
[9] Aus den gleichen Gründen ist ein Einschränken des Suchbereichs in der Auflösungspyramide von RGB-Verfahren durch die Tiefeninformation nur eingeschränkt möglich.

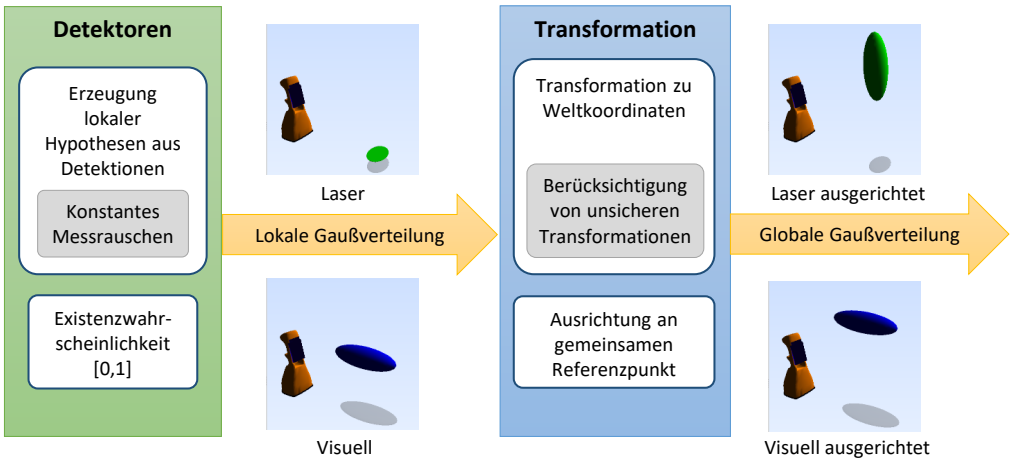


Abbildung 3.13.: Schematischer Ablauf der Erzeugung von lokalen Hypothesen und der Transformation in globale Weltkoordinaten. Erläuterung der einzelnen Verarbeitungsschritte im Text.

3.5. Generierung von 3D-Hypothesen

Bei der Spezifikation der Systemarchitektur (Abschnitt 2.3) wurde definiert, dass die Hypothesen aller Detektoren über eine gemeinsame Schnittstelle an den Personentracker übergeben werden. Diese Hypothesen sollen dabei in Weltkoordinaten vorliegen, um die Fusion im Tracker zu erleichtern. Tracking in Weltkoordinaten ermöglicht intuitive Bewegungsmodelle bei einem sich bewegenden Roboter und erfordert keine Eigenbewegungskompensation. Dies wird in der vorliegenden Architektur durch eine bekannte Roboterpose ermöglicht, welche durch Monte-Carlo oder SLAM Algorithmen (Einhorn u. a. 2014) geschätzt wird.
Dieser Abschnitt beschreibt den Aufbau der Hypothesen und die Transformationen aus den verschiedenen Detektoren in Weltkoordinaten. Letzteres bezieht zusätzlich die Unsicherheit über die aktuelle Roboterpose ein. Diese Abfolge ist schematisch in Abbildung 3.13 dargestellt. Ein wesentlicher Vorzug dieser Herangehensweise ist, dass der Tracker mithilfe der Ausrichtung auf ein gemeinsames Basissystem allgemeingültig mit beliebigen Detektionsmodulen zusammenarbeiten kann.

3.5.1. 3D-Hypothesen

In der entwickelten Architektur werden Personen mittels einer Kombination aus Gaußhypothesen und diskreten Verteilungen beschrieben. Diese beschreiben die 3D-Position, Existenzwahrscheinlichkeit und weitere Eigenschaften ei-

ner Person in der Umgebung.

Definition 3.1: Personenhypothese

Eine Personenhypothese sei als Normalverteilung über der Position im kartesischen Raum definiert:

$$h(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\} \tag{3.1}$$

Dabei bezeichnet der 3-dimensionale Mittelwert:

$$\boldsymbol{\mu} = \mathbb{E}[\mathbf{x}] = (x, y, z)^T$$

die Position der Hypothese im Raum und die 3×3 Matrix $\boldsymbol{\Sigma}$ kennzeichnet die zugehörige Kovarianz:

$$\boldsymbol{\Sigma} = \mathrm{cov}[\mathbf{x}]$$

Die Personenhypothese dient als Repräsentationsform der Detektionen aller Detektionsmodule. Weiterhin wird sie für die Darstellung und Weiterverarbeitung der gefilterten Zustände des Personentrackers verwendet. Für jede Hypothese wird zusätzlich eine Existenzwahrscheinlichkeit bestimmt:

Definition 3.2: Existenzwahrscheinlichkeit

Die Existenzwahrscheinlichkeit definiert sich als Bernoulli-Verteilung mit:

$$\begin{aligned} p(H = Person) &= p \\ p(H = Objekt) &= 1 - p \quad \text{, mit } p \in]0, 1[\end{aligned} \tag{3.2}$$

Sie bezeichnet die Wahrscheinlichkeit, dass es sich bei dem detektierten oder getrackten Objekt H tatsächlich um eine Person handelt.

Die Existenzwahrscheinlichkeit wird mittels eines Naive-Bayes-Klassifikators geschätzt (Abschnitt 4.8). Sie unterstützt die Entstehung und Beendigung von Personenspuren im Personentracker und kann für nachfolgende Module (z. B. Dialog und Folgeverhalten) als Konfidenzwert zur Auswahl sicherer Hypothesen genutzt werden.

Neben dieser Information können noch zusätzliche Eigenschaften, wie Geschwindigkeit, Bewegungs- oder Blickrichtung in der multivariaten Gaußverteilung der Hypothese repräsentiert werden. Geschwindigkeit und Bewegungsrichtung werden von den, in dieser Arbeit verwendeten, Detektoren nicht ge-

schätzt, sondern sind ein Resultat der Systemmodelle im Personentracker (Abschnitt 4.5). Das Schätzen der Blickrichtung ist nicht Gegenstand dieser Arbeit. Die Architektur wurde jedoch erfolgreich zum Nutzen der erkannten Blickrichtung von Detektoren (Weinrich u. a. 2012), als auch zum Tracken der Blickrichtung auf Basis von Bewegungsmodellen im Personentracker eingesetzt (Weinrich u. a. 2013b; Weinrich 2016).

3.5.2. Messmodell

Das Mess- oder Sensormodell beschreibt die Wahrscheinlichkeitsverteilung einer Beobachtung $\mathbf{z}$ unter der Bedingung eines Zustandes $\mathbf{x}$:

$$p(\mathbf{z} \mid \mathbf{x}) \tag{3.3}$$

Das Messmodell findet in der Bayes-Filterung (Algorithmus 4.1) im Personentracker Anwendung. An dieser Stelle werden die Besonderheiten der einheitlichen Schnittstelle der Detektoren in der Architektur beschrieben (Abschnitt 2.3.2). Die Messunsicherheit wird in dieser Arbeit einerseits durch die Kovarianzmatrix der Gaußverteilung (Definition 3.1) und durch die Existenzverteilung (Definition 3.2) ausgedrückt.
Das Sensormodell wird demnach in den Detektionsmodulen selbst implementiert, welche aus den gemessenen Sensordaten die entsprechenden Wahrscheinlichkeitsverteilungen erzeugen. Durch diese Vorgehensweise wird das Messrauschen unabhängig vom aktuellen Zustand $\mathbf{x}$. Die Architektur erlaubt es, dass ein Detektor ein dynamisches Messrauschen abhängig von der Detektion im Sensorraum, beziehungsweise der transformierten Position in Weltkoordinaten angibt. So kann die Kovarianz der Gaußverteilung beispielsweise mit zunehmender Entfernung erhöht werden. Eine Modellierung der Abhängigkeit von anderen Größen im Zustandsraum des Personentrackers, welche im Detektor nicht bekannt sind (z. B. der Geschwindigkeit), ist allerdings nicht möglich. Diese Einschränkung stellte sich für das vorliegende Anwendungsszenario als nicht relevant heraus (Kapitel 7).
Die Detektoren dieser Arbeit verwenden konstante, experimentell bestimmte Werte für die Varianzen der Gaußverteilung. Zwar wird die Schätzung dadurch mit zunehmender Entfernung ungenauer, in der Praxis fällt dieser Unterschied jedoch minimal aus, da der Erfassungsbereich des Roboters im Bereich von ca. vier Metern liegt. Dies steht im krassen Gegensatz zu Fahrerassistenzsystemen, welche mit Entfernungen von bis zu 200 m umgehen (Schubert u. a. 2010). In letztgenanntem Fall wirkt sich eine beispielhafte Abweichung von einem Pixel einer detektieren Bounding-Box signifikant auf die geschätzte Position aus.
Die nachfolgenden Abschnitte beschreiben die Generierung der Gaußvertei-

lungen mit zugehöriger Kovarianz, während sich die Existenzwahrscheinlichkeit jeweils aus der Richtig-positiv- und Falsch-positiv-Rate des Detektors auf passenden Datensätzen ergibt und in Abschnitt 4.8 erläutert wird.

3.5.3. Generierung aus Laserdetektionen

In Abschnitt 3.2.2 wurde gezeigt, wie Segmente im Laserscan zu Beinen oder Hintergrund klassifiziert werden. Aus den Schwerpunkten zweier nah beieinanderliegender Beinsegmente $S_i = (x_i, y_i)$ und $S_j = (x_j, y_j)$ ergibt sich der Mittelwert einer Personenhypothese h_k relativ zum Lasersensor zu:

$$\boldsymbol{\mu}_k = \left(\frac{x_i + x_j}{2}, \frac{y_i + y_j}{2}, 0 \right) \tag{3.4}$$

Da alle erkannten Hypothesen auf der Ebene des Laserscanners liegen, wird $z_k = 0$ gesetzt. Die vorläufige Kovarianzmatrix ergibt sich zu:

$$\hat{\boldsymbol{\Sigma}}_k = \begin{pmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_\phi^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{pmatrix} \tag{3.5}$$

Hierbei bezeichnet σ_r^2 die Varianz der Messung in der Tiefe und σ_ϕ^2 die Varianz senkrecht hierzu in der Scanebene. σ_z^2 definiert die Varianz senkrecht zur Scanebene. Da der Messwinkel ϕ für jede Detektion unterschiedlich ist, muss $\hat{\boldsymbol{\Sigma}}_k$ rotiert werden, sodass wie angegeben σ_r^2 die Unsicherheit entlang des, durch die Punkte $P_0 = (0, 0)$ und $P_k = (r_k, \phi_k)$ bzw. $\boldsymbol{\mu}_k$ definierten, Messstrahls liegt (Abbildung A.2).
Die intrinsische Rotationsmatrix $\mathbf{R}$ ergibt sich aus drei Rotationen um die Koordinatenachsen:

$$\mathbf{R} = \mathbf{R}_z(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_x(\gamma) \qquad \text{mit} \quad \mathbf{R}_x(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{pmatrix} \tag{3.6}$$

$$\mathbf{R}_y(\beta) = \begin{pmatrix} \cos(\beta) & 0 & sin(\beta) \\ 0 & 1 & 0 \\ -sin(\beta) & 0 & \cos(\beta) \end{pmatrix} \qquad \mathbf{R}_z(\alpha) = \begin{pmatrix} \cos(\alpha) & -sin(\alpha) & 0 \\ sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

wobei α, β und γ den yaw, pitch, roll Winkeln[10] entsprechen (Arfken u. a. 2008). Im Falle des Lasersensors gilt weiterhin: $\alpha = \phi_k$ und $\beta = \gamma = 0$.

[10] yaw, pitch, roll entsprechen einer Drehung um die z, y, beziehungsweise x-Achse.

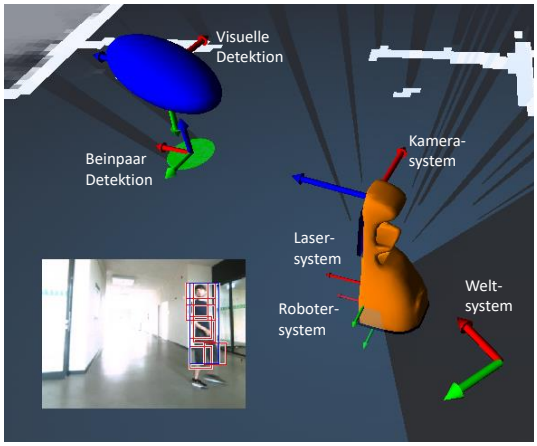


Abbildung 3.14.: Koordinatensysteme. Die Farben RGB der Pfeile entsprechen den x, y, z-Achsen. Die Roboterpose befindet sich im Weltkoordinatensystem und spannt ein eigenes Koordinatensystem auf. Relativ dazu sind das Laser- und Kamerakoordinatensystem definiert. Beinpaardetektion mit Kovarianzellipsoid in grün und visuelle Detektion mit großer Varianz in Richtung zur Kamera in Blau in ihren jeweiligen Sensorkoordinatensystemen. Kamerabild unten links.

Die rotierte Kovarianz der Hypothese (im Sensorraum) ergibt sich mit:

$$\mathbf{\Sigma}_k = \mathbf{R}\hat{\mathbf{\Sigma}}_k\mathbf{R}^T \tag{3.7}$$

Da die Segmentmittelpunkte S_i und S_j nicht exakt den Mittelpunkten der menschlichen Beine entsprechen und sich die Beine einer Person nicht immer exakt im Zentrum der Person befinden (z. B. sitzende Pose), werden σ_r^2 und σ_ϕ^2 auf einen empirisch ermittelten Wert von 0.2 m gesetzt. Die Unsicherheit in der Höhe wird durch σ_z^2 beschrieben und auf den Epsilon-Wert $\epsilon = 1 \times 10^{-6}$ gesetzt. Abbildung 3.14 zeigt neben den verwendeten Koordinatensystemen eine beispielhafte Beinpaar-Hypothese, die sich im Laserkoordinatensystem befinden. Die Hypothesen müssen zur Weiterverwendung in Weltkoordinaten transformiert werden (Abschnitt 3.5.5).

3.5.4. Generierung aus Bilddetektionen

Zur Generierung einer 3D Hypothese aus einer Bilddetektion (typischerweise eine Bounding-Box) wird diese in eine Gaußverteilung im Kamerakoordinatensystem umgewandelt. Die Abbildung von Personen auf die Bildebene nach dem Lochkameramodell ist in Anhang A.4.1 beschrieben. Der Mittelpunkt des Rechtecks P_i wird mittels inverser Kameraprojektion und Kenntnis der Perso-

nenbreite (Anhang A.4.2) auf einen Punkt P_k in Kamerakoordinaten transformiert. Dieser Punkt entspricht dem Mittelpunkt der Hypothese:

$$\boldsymbol{\mu}_l = P_k \tag{3.8}$$

Die vorläufige Kovarianzmatrix ergibt sich zu:

$$\hat{\boldsymbol{\Sigma}}_k = \begin{pmatrix} \sigma_{x_k}^2 & 0 & 0 \\ 0 & \sigma_{y_k}^2 & 0 \\ 0 & 0 & \sigma_{z_k}^2 \end{pmatrix} \tag{3.9}$$

wobei σ_i^2 der jeweiligen Varianz in Richtung der Achse im Kamerakoordinatensystem entspricht. Die gedrehte Kovarianzmatrix ergibt sich analog nach Gleichungen (3.6) und (3.7), wobei α, β, γ (yaw, pitch, roll) mit:

$$\begin{aligned} &\alpha = 0 && \text{Drehung um z-Achse} \\ &\beta = \arctan(\frac{x_k}{z_k}) && \text{Drehung um y-Achse} \\ &\gamma = \arctan(-\frac{y_k}{z_k}) && \text{Drehung um x-Achse} \end{aligned} \tag{3.10}$$

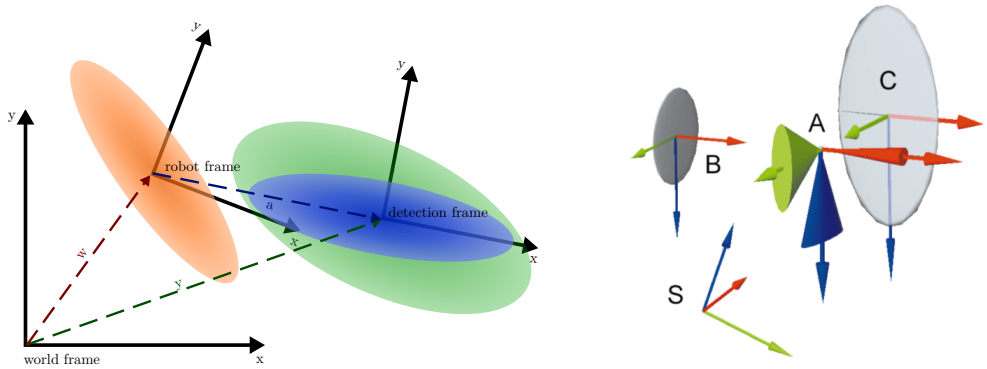
die Rotation des Strahls im Kamerakoordinatensystem angeben. Die Werte für $\sigma_{x_k}^2$, $\sigma_{y_k}^2$ und $\sigma_{z_k}^2$ werden empirisch bestimmt, wobei für $\sigma_{z_k}^2$ eine relativ große Varianz gesetzt wird, da sich die Entfernung (Tiefe) anhand der Breite der Bounding-Box nur unsicher bestimmen lässt. Eine beispielhafte visuelle Detektion im Kamerakoordinatensystem ist in Abbildung 3.14 zu finden.

3.5.5. Transformation in globale Weltkoordinaten

Auf einem Robotersystem existieren gewöhnlich verschiedene lokale Koordinatensysteme, z. B. die Pose des Roboters in der Welt, die Position des Laserscanners und die Position und Orientierung der Kamera. In dieser Arbeit wird für die Verwaltung der Koordinatensysteme und ihrer Beziehungen zueinander auf einen Transformationsbaum zurückgegriffen (Einhorn u. a. 2012). Dieser speichert, wie bei einem Szenengraph, für jedes Koordinatensystem die aktuelle Transformation zum jeweiligen Elternknoten. Diese werden als euklidische Transformationen (engl. *rigid transform*) durch eine Translation und Rotation (Isometrie) beschrieben:

$$T(\mathbf{v}) = \mathbf{R}\mathbf{v} + \mathbf{t} \tag{3.11}$$

Dabei beschreibt $T(\mathbf{v})$ den transformierten Vektor $\mathbf{v} = (x, y, z, \alpha, \beta, \gamma)^T$, $\mathbf{t}$ die



(a) 2D Positionsunsicherheit **(b)** 3D Rotationsunsicherheit

Abbildung 3.15.: Covariance Error Propagation. (a) Kovarianz der Transformation **y** aus zwei unsicheren Transformationen **w** und **a**. Die Unsicherheit der Detektion (blau) und der Roboterpose (orange) wird zur Kovarianz in Weltkoordinaten (grün) propagiert. (b) Pose A mit Unsicherheit in der Rotation bezüglich des Weltkoordinatensystems S (Kegel an den Achsen). Obwohl Posen B und C keine Unsicherheit bezüglich Pose A haben, erhöht sich deren Positionsunsicherheit bezüglich S, da die Orientierung von A unsicher ist. Quelle: Hoff u. a. (2000)

Translation und Matrix **R** die Rotation (Galarza u. a. 2007). Mithilfe von mehreren hintereinander angewendeten Transformationen werden die Hypothesen der Detektoren aus ihren lokalen Koordinatensystemen in ein globales Weltkoordinatensystem transformiert[11]. Bezeichne $\mathbf{T}_D^R$ die homogene Transformation einer Detektion bezüglich des Roboterkoordinatensystems und $\mathbf{T}_R^W$ die Transformation des Roboters bezüglich des Weltkoordinatensystems, dann gilt für die Transformation der Detektion bezüglich des Weltkoordinatensystems $\mathbf{T}_D^W$ nach Craig (2005):

$$\mathbf{T}_D^W = \mathbf{T}_R^W \mathbf{T}_D^R \tag{3.12}$$

Unsichere Transformationen

Die Transformation aus Gleichung (3.11) überführt den Mittelwert und die Kovarianz der Hypothesen in das neue Koordinatensystem. Am Beispiel der

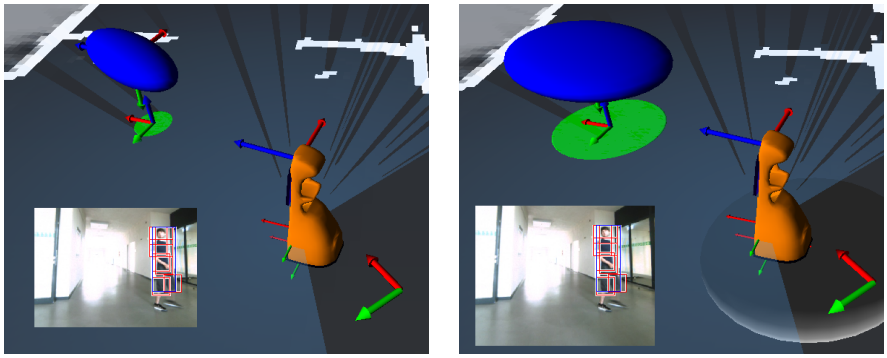
[11] Im eingesetzten Framework (Einhorn u. a. 2012) werden die Rotationen zur Effizienzsteigerung und Vermeidung des Gimbal Locks als Quaternionen gespeichert. Der Gimbal Lock bezeichnet bei Transformationen mit Eulerwinkeln das Zusammenfallen der Achsen der ersten und der dritten Drehung. Die Kovarianzmatrix in Quaternionenform ist demnach 7-dimensional.

Roboterpose soll der Einfluss von unsicheren Transformationen erläutert werden[12]. Wenn die Pose des Roboters in der Welt eindeutig bekannt ist, führt die Transformation zu einer korrekten Überführung der Hypothesen in das Weltkoordinatensystem. Da der Roboter seine eigene Pose jedoch selbst mittels Monte Carlo Lokalisation (König u. a. 2005) oder SLAM (Einhorn u. a. 2014) schätzt, ist diese mit Unsicherheiten behaftet. Der Transformationsbaum speichert daher für jede Transformation nicht nur die Translation $\mathbf{t}$ und Rotation $\mathbf{R}$, sondern auch die zugehörigen Unsicherheiten in einer Kovarianzmatrix. Die Transformation lässt sich demnach ebenfalls als Normalverteilung interpretieren.
Da sich die lokalen Hypothesen im Kamera-, Laser- und Roboterkoordinatensystem mit dem Roboter bewegen, muss bei der Transformation in das Weltkoordinatensystem auch die Unsicherheit der Roboterpose berücksichtigt werden. Im Rahmen dieser Arbeit wird hierfür die Übertragung der Kovarianzfehler (engl. *Covariance Error Propagation*) genutzt.
In Abbildung 3.15(a) ist dies schematisch dargestellt. Darin bezeichnet Transformation $\mathbf{w}$ die Roboterpose im Weltkoordinatensystem mit zugehöriger Unsicherheit $\mathbf{\Sigma_w}$, die durch die orange Kovarianzellipse ausdrückt wird. Das Koordinatensystem des Roboters ist mit „robot frame“ bezeichnet. Erzeugt ein visuelles Erkennungsmodul eine Detektion (blau) besitzt diese eine hohe Varianz in der Blickrichtung der Kamera (x-Richtung des Roboters). Diese ist durch eine Transformation $\mathbf{a}$, deren Position und Unsicherheit $\mathbf{\Sigma_a}$ im Roboterkoordinatensystem angegeben ist, beschrieben. Die Kovarianz der Detektion in Weltkoordinaten (grün) vereinigt beide Kovarianzen und wird mittels Covariance Error Propagation (Hoff u. a. 2000; Simpson u. a. 2011) berechnet:

$$\mathbf{\Sigma_y} = \mathbf{J_a}\mathbf{\Sigma_a}\mathbf{J_a}^T + \mathbf{J_w}\mathbf{\Sigma_w}\mathbf{J_w}^T \tag{3.13}$$

Hierbei bezeichnet $\mathbf{\Sigma_y}$ die Kovarianz der verketteten Transformation $\mathbf{y} = \mathbf{g}(\mathbf{w}, \mathbf{a}) = \mathbf{w} \cdot \mathbf{a}$, und $\mathbf{\Sigma_a}$, $\mathbf{\Sigma_w}$ bezeichnen die jeweils die Kovarianzen der Transformationen $\mathbf{a}$ und $\mathbf{w}$. Die Jacobi Matrizen sind durch $\mathbf{J_a} = \partial\mathbf{g}/\partial\mathbf{a}$ und $\mathbf{J_w} = \partial\mathbf{g}/\partial\mathbf{w}$ gegeben. Zur besseren Anschaulichkeit visualisiert Abbildung 3.15(a) nur die Positionsunsicherheit in 2D. Ein Beispiel für die Auswirkungen der Rotationsunsicherheit in 3D ist in Abbildung 3.15(b) gegeben (Hoff u. a. 2000). Hier sei Pose A mit einer Unsicherheit in der Rotation bezüglich des Weltkoordinatensystems S behaftet. Posen B und C seien mit einer Transformation ohne Unsicherheit bezüglich A gegeben. Die Posen B und C bezüglich S sind mit $\mathbf{T}_B^S = \mathbf{T}_A^S\mathbf{T}_B^A$ und $\mathbf{T}_C^S = \mathbf{T}_A^S\mathbf{T}_C^A$ gegeben. Obwohl die Positionen von B und C bezüglich A exakt bekannt sind, haben diese eine

[12] Gleiches gilt entsprechend für den Einfluss einer unsicheren Sensorpositionierung am Roboter.



(a) Sichere Roboterpose **(b)** Unsichere Roboterpose

Abbildung 3.16.: Kovarianzellipsoiden einer Laser- und visuellen Detektion mit (a) sicherer Roboterpose und mit (b) unsicherer Roboterpose. Durch die Kovarianzfehlerübertragung vergrößern sich die Kovarianzen der Detektionen bezüglich des Weltkoordinatensystems.

relativ hohe Unsicherheit in der Position bezüglich S, da die Orientierung von A unsicher ist. Die Auswirkung einer unsicheren Position auf die Hypothesen der Detektoren ist in Abbildung 3.16 dargestellt.

3.5.6. Ausrichtung von Hypothesen

Viele Detektionsmodule erkennen Personen anhand unterschiedlicher Körperpartien, z. B. Beinpaare, Gesicht, Kopf-Schulter- und Ganzkörperkontur. Zur Fusion im Tracker werden die Hypothesen der Detektionsmodule auf einen gemeinsamen Referenzpunkt ausgerichtet. In dieser Arbeit wird der Kopf einer Person verwendet.
Dazu wird der Mittelwert jeder Hypothese entlang der vertikalen z-Achse zur erwarteten Kopfposition geschoben. Weiterhin wird die Kovarianz entsprechend der Unsicherheit der Kopfposition relativ zur detektieren Körperpartie erhöht. Abbildung 3.17 visualisiert diesen Vorgang. Eine visuelle Hypothese (blaues Kovarianzellipsoid), die aus dem Mittelpunkt einer Bounding-Box um die gesamte Person entstanden ist, wird daher um die Entfernung Personenmittelpunkt-Kopf nach oben geschoben. Die Kovarianz bleibt unverändert (cyanes Ellipsoid). Laserdetektionen (grünes Ellipsoid) werden auf die durchschnittliche Höhe eines Personenkopfes geschoben und die Kovarianz in vertikaler Richtung erhöht (gelbes Ellipsoid). Diese zusätzliche Unsicherheit beschreibt die verschiedenen Kopf-Bein Entfernungen von sitzenden, stehenden sowie großen und kleinen Personen.

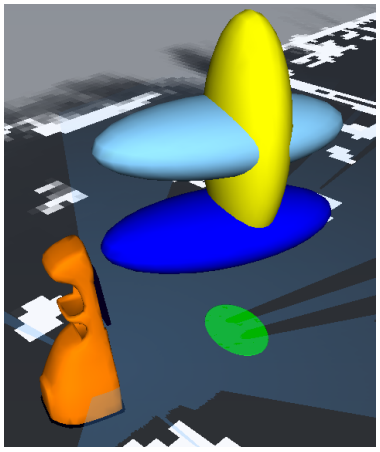


Abbildung 3.17: Die Hypothesen unterschiedlicher Detektoren werden an einem gemeinsamen Referenzpunkt ausgerichtet (Kopf der Person). Eine Laserhypothese (Grün) wird auf die Durchschnittsgröße einer Person geschoben und die Kovarianz in z-Richtung erhöht (Gelb). Eine visuelle Hypothese auf Körperhöhe (Blau) wird auf Kopfhöhe geschoben (Cyan). Die ausgerichteten globalen Hypothesen (Gelb, Cyan) werden an den Personentracker weitergegeben.

3.6. Experimentelle Untersuchungen

Die Evaluation der Detektoren erfolgt im Zuge der Evaluierung des Personentrackers in Abschnitt 4.10.3.

3.7. Diskussion und Fazit

Dieses Kapitel systematisierte die bekannten abstandsbasierten, visuellen und tiefenbasierten Ansätze für mobile Systeme. Dabei wurden Verfahren, die auf dem mobilen Assistenzroboter eingesetzt werden, näher erläutert. Im Rahmen der Dissertation wurden Erweiterungen und Verbesserungen einiger Verfahren durchgeführt.
Das in vielen aktuellen Projekten eingesetzte Verfahren zur Beindetektion von Arras u. a. (2007) wurde in der Arbeit um binäre Entscheidungsbäume erweitert, was die Klassifikationsgüte beträchtlich erhöht. Erweiterungen, die in der Arbeit nicht betrachtet wurden, umfassen weitere Merkmale und Klassifikationskaskaden (Weinrich u. a. 2014b,a) beziehungsweise der Einsatz von Laserarrays und 3D Lasersensoren (Spinello u. a. 2011a; Mozos u. a. 2010).
Die Einschränkung der Suchbereiche des Gesichtsdetektors (Viola u. a. 2002) und der HOG Detektoren (Dalal u. a. 2005; Ferrari u. a. 2008) mittels Region of Interest beziehungsweise Ground Plane Constraint beschleunigen die Verfahren und reduzieren Falsch-positive. Der Gesichtsdetektor und die HOG Verfahren sind einzeln in vielen Projekten zu finden (Choi u. a. 2013; Jafari u. a. 2014), werden jedoch selten kombiniert. Das eingesetzte Verfahren von Dollar u. a. (2010) erlaubt die echtzeitfähige Detektion von stehenden Personen. Beschleunigungen, die eine GPU voraussetzen, konnten nicht verwendet werden (Benenson u. a. 2012). Der Einsatz eines Deformable Part Models (P. F.

Felzenszwalb u. a. 2010) ermöglicht die Wahrnehmung von Personen in unterschiedlichen Posen und unter Verdeckungen. Erst durch eine Transformation in den Fourier Raum nach Dubout u. a. (2012) und Anpassungen lässt sich das Verfahren auf einem mobilen Roboter echtzeitfähig einsetzen.
3D Verfahren auf einer Tiefenkamera wurden im Rahmen der Dissertation untersucht (Hordern u. a. 2010; Spinello u. a. 2011b; Hegger u. a. 2013), eignen sich jedoch aufgrund von mangelnder Detektionsleistung und Hardwareeinschränkungen nicht für den Einsatz im Anwendungsszenario. Neuere Verfahren auf einem Tiefensensor mit großem Sensorfeld (Kinect 2) können zukünftig von Vorteil für die Personendetektion sein (Jafari u. a. 2014).
In den letzten Jahren stellten sich Deep Learning Verfahren als der treibende Faktor für Fortschritte in der Personendetektion heraus. Zurzeit fehlen für den Durchbruch der Verfahren auf mobilen System jedoch entweder Strom sparende, leistungsfähige GPUs oder Optimierungen der Verfahren für CPUs. Die Verfahren sind daher nicht Bestandteil dieser Arbeit, können aber einfach als neuer Detektor integriert werden.
Möglich wird dies durch eine gemeinsame Schnittstelle für die Hypothesen der Detektoren zum Tracker. Das Messmodell wird in den jeweiligen Detektor verlagert und ist damit unabhängig vom aktuellen Zustand im Personentracker. Dazu erzeugt jeder Detektor zunächst lokale 3D Gaußverteilungen und liefert seine spezifische Richtig- und Falsch-positiv-Rate. Diese werden anschließend unter Berücksichtigung von unsicheren Transformationen (z. B. der geschätzten Roboterpose) in Weltkoordinaten transformiert. Eine anschließende Ausrichtung auf einen gemeinsamen Referenzpunkt bereitet die Hypothesen für den Einsatz im Personentracker vor. Zukünftige Arbeiten können die derzeit empirisch festgelegten Unsicherheiten der Detektoren in der Lokalisierung aus Trainingsdaten lernen.

4. Personentracking

Im vorangegangenen Kapitel wurden die verwendeten Detektionsmodule und die Generierung von 3D Hypothesen beschrieben. In diesem Kapitel wird die Arbeitsweise des Personentrackers erläutert, welcher die Hypothesen fusioniert und raumzeitlich filtert. Zunächst werden in Abschnitt 4.2 bekannte Ansätze systematisiert. Anschließend gibt Abschnitt 4.3 einen konzeptionellen Überblick über den entwickelten Personentracker. Die Abschnitte 4.4 und 4.5 beschreiben die rekursive Zustandsschätzung und die Systemmodelle. Deren Besonderheiten in der softwaretechnischen Umsetzung werden in Abschnitt 4.6 erläutert. Die Abschnitte 4.7 bis 4.9 beschreiben weitere wichtige Teilfunktionalitäten wie die Schätzung von Existenzwahrscheinlichkeiten, Posenschätzung und die Generierung und Beendigung von Hypothesentracks.

4.1. Einleitung

Mobile Roboter sollen den Nutzer möglichst unabhängig von dessen Position und Pose wahrnehmen. Aufgrund der Mobilität des Nutzers kommt es jedoch zwangsweise zu Verdeckungen, z. B. durch Möbel oder Gegenstände im Raum. Zusätzlich erschweren verschiedene Nutzerposen und Ansichten die Wahrnehmung. Verdeckungen und die Posenvarianz führen zu fehlenden Hypothesen der Detektionsmodule (Falsch-negative).
Trackingverfahren können diese kurzfristigen Ausfälle der Detektoren durch die Prädiktion der Personenbewegung kompensieren. Die prädizierte Positionsschätzung wird durch die Fusion von Hypothesen unterschiedlicher Module korrigiert und verbessert. Falsch-positive einzelner Detektoren können ausgeschlossen werden. Prädiktion und Fusion beruhen dabei auf dem Prinzip der Bayes-Filterung und einem Systemmodell. Über Beziehungen zwischen Beobachtungs- und Prädiktionsmodellen können Eigenschaften, z. B. die Geschwindigkeit von Personen geschätzt werden, die nicht direkt beobachtet werden. Bei der Bayes-Filterung handelt es sich um ein probabilistisches Verfahren zur Zustandsschätzung. Probabilistische Verfahren modellieren die inhärente Unsicherheit des Einsatzszenarios. Die Unsicherheiten treten dabei unter anderem in der Umgebung, den Sensoren, dem Roboter, den Modellen und der Berechnung auf (Thrun u. a. 2005). Als Beispiel sind Sensoren in ihrer Reich-

weite, Messgenauigkeit (Rauschverhalten) und Messprinzip (Kameras können nicht durch Wände schauen) beschränkt. Weiterhin sind die angenommenen Modelle von Roboter und Personen niemals exakt modellier- und berechenbar.

4.2. Systematisierung bekannter Trackingansätze

Personentracker sind fester Bestandteil mobiler Systeme. Viele aktuelle Trackingsysteme stammen aus der Fahrerassistenz, dem Fußgängertracking, dem rein visuellen Tracking oder direkt aus der mobilen Servicerobotik. Die Algorithmen sind jeweils von den verwendeten Sensoren und Detektoren abhängig.

4.2.1. Automobiler Bereich

Die am häufigsten im automobilen Bereich eingesetzten Sensoriken umfassen einen Radar- bzw. Lasersensor und eine Stereo- oder Mono-RGB-Kamera. Bajracharya u. a. (2009) setzen einen heuristischen Tracker im Bildraum ein, der Personentracks mit der nächstgelegenen Bounding-Box fortführt. Giebel u. a. (2004) kombinieren Kontur, Textur und Tiefe in einem Partikel-Filter Framework, um Fußgänger visuell aus einem Fahrzeug heraus zu tracken. Gavrila u. a. (2006) erzeugen mittels Stereo-Kamera Regions of Interest für einen Template-Detektor, dessen Detektionen mit einem α-β-Tracker gefiltert werden. Spinello u. a. (2010a) beschränken den Suchraum eines Implicit Shape Models (Leibe u. a. 2003) mittels Laserdaten und trackt Personen mit mehreren Kalman-Filtern im Laserkoordinatensystem. Leibe u. a. (2008) und Ess u. a. (2008) wandeln Bildhypothesen mittels Schätzung der Kamerapose in 3D Weltkoordinaten und tracken diese mit einem komplexen globalen Smoothing Algorithmus bzw. einem Graphischen Modell, dass Verdeckungen inferiert. Allerdings arbeiten die vier zuletzt genannten Verfahren, trotz Einsatz leistungsfähiger CPUs und GPUs, nur mit 1-3 Frames pro Sekunde. Unter Verwendung von einer dedizierten GPU, einer Stereo-Kamera erweitern Mitzel u. a. (2011) das Trackingverfahren von Leibe u. a. (2008) zu einem echtzeitfähigen Personentracking auf Basis von Kalmanfiltern. Dabei wird jedoch nur die benötigte Rechenzeit des Trackingsystems betrachtet, ohne die Berechnungszeit der Module für die Stereoberechnung und die Lokalisation des Fahrzeugs einzubeziehen.

4.2.2. Visuelles Personentracking

Rein visuelle Personentracker erzeugen Spuren von Bounding-Boxen im Bild einer Kamera. Andriluka u. a. (2008) nutzen ein aufwendiges Graphisches Modell, um Personen und deren Artikulationen unter häufigen Verdeckungen zu

tracken. Das Verfahren von Breitenstein u. a. (2011) erreicht beeindruckende Trackingergebnisse unter Nutzung einer unkalibrierten Kamera und einem Partikel-Filter. Beide Verfahren sind jedoch weit von einer echtzeitfähigen Implementierung entfernt. Klein u. a. (2010) nutzen einen Partikel-Filter und ein adaptives Beobachtungsmodell, um beliebige Objekte im Bild zu tracken. Allerdings muss der Tracker manuell initialisiert werden. Choi u. a. (2013) stellen ein generelles Partikel-Filter basiertes Trackingsystem vor, welches visuelle Detektoren, darunter Gesichts-, HOG, DPM und Tiefen-Templates, im RGB- oder Tiefenbild nutzt, um Personen in Weltkoordinaten zu tracken und die Kamerapose automatisch anhand von statischen Bildfeatures zu schätzen. Das Verfahren erreicht mit ca. 2-5 fps keine Echtzeitfähigkeit. Jafari u. a. (2014) nutzen ein Tiefentemplate und ein GPU-HOG, um Personen mit dem in Leibe u. a. (2008) vorgestellten Verfahren zu tracken. Dieses passt mehrere Trajektorien mittels EKF Smoothing an die vorhandenen Hypothesen der Detektoren an und selektiert die am besten Passenden. Das Verfahren läuft in Echtzeit.
Verfahren, die Personentracker ausschließlich auf Basis von abstandsbasierten Daten implementieren, sind in Arras u. a. (2008), Lee u. a. (2008) und Kaestner u. a. (2012) zu finden.
Visuelle Trackingverfahren erzielen sehr gute Ergebnisse und sind teilweise echtzeitfähig. Im Gegensatz zu dieser Arbeit werden jedoch keine unterschiedlichen Sensormodalitäten allgemeingültig kombiniert. Eine umfangreiche Bestandsaufnahme verschiedener visueller Trackingverfahren findet sich in Yilmaz u. a. (2006) und Smeulders u. a. (2014).

4.2.3. Personentracking auf mobilen Robotern

Mobile Roboter sind in der Regel mit relativ schlechter Rechenleistung und Sensorik ausgestattet. Daher werden effiziente, approximative Trackingalgorithmen, wie Extended- und Unscented Kalman-Filter, und einfache Detektionsverfahren eingesetzt (Bellotto u. a. 2009). Häufig wird von einer aufrechten, stehenden Pose der Personen im Kamerabild und vorhanden Beinpaaren im Laserscan ausgegangen (Bellotto u. a. 2010). Der Personentracker des PR2 (Pantofaru 2011) initialisiert Personentracks mit Gesichtsdetektionen sowie 3D Verifikation und trackt diese anschließend mit Beinpaardetektionen und Höheneinschränkung im Tiefenbild. Bellotto u. a. (2010) integrieren Gesichts-, Kleidung- und Beinpaardetektionen in einem sequenziellen Korrekturschritt. Das bedeutet, dass die einzelnen Detektoren fest integriert sind und sukzessive nach Detektionen abgefragt werden. B. Wu u. a. (2007) und Liu u. a. (2010) tracken Personen im Kamerabild eines Roboters unter Nutzung des (adaptiven) Meanshift Algorithmus, nachdem die Person mit Part-Kantenfiltern detektiert wurde. Xudong u. a. (2008) kombinieren eine Laserdetektion mit einem visu-

ellen Farbtracking unter Verwendung eines Unscented Partikel-Filters.
Einige Verfahren nutzen ein rein visuelles Personentracking auf mobilen Robotern. Brèthes u. a. (2008) verwenden Farbe, Kontur, Textur, Tiefe und Bewegung und einen Partikel-Filter, während Mitzel u. a. (2010) die Zeit zwischen zwei HOG Detektionen mit einem Level-Set Tracker überbrücken, dessen kurze Tracks von einem Kalman-Filter fusioniert werden. Cielniak u. a. (2010) verwenden einen Konturtracker auf einer Wärmebildkamera. Basso u. a. (2013) tracken Personen mittels Unscented Kalman-Filtern in den RGB-D Daten der Kinect unter Verwendung von tiefenbasiertem Clustering, HOG Detektionen und online Klassifikatoren zur Wiedererkennung. Neben dem eigentlichen Personentracking beschäftigen sich Luber u. a. (2011) mit personenspezifischen Modellen zur Entstehung und Löschung von Tracks, Falsch-positiven und Verdeckungen.

4.2.4. Bewertung

Die vorgestellten Verfahren zum Personentracking auf mobilen Systemen verwenden am häufigsten Bayes-Filter Algorithmen. Speziell kommen Kalman-Filter (Kalman 1960; S. J. Julier u. a. 2000) und Partikel Filter zum Einsatz (Isard u. a. 1998; Rubinstein u. a. 2008). Multi-Hypothesen-Tracker, die verschiedene Varianten der Beobachtungszuordnung verfolgen sind im mobilen Bereich eher selten zu finden (Lau u. a. 2009; Mucientes u. a. 2006). Viele Verfahren erzielen sehr gute offline Ergebnisse oder nutzen Sensoren und Hardware, wie Stereokamera und GPUs, die auf dem in dieser Arbeit verwendeten Roboter nicht verfügbar sind. Das Tracking selbst ist jedoch häufig stark auf die gegebenen Sensoren und Detektoren zugeschnitten. In dieser Arbeit wird ein allgemeines Konzept vorgestellt, das es erlaubt, die Hypothesen von verschiedenen heterogenen Detektoren in den Personentracker zu integrieren.

4.3. Konzeptioneller Aufbau

Dieser Abschnitt beschreibt den im Rahmen dieser Arbeit entwickelten Personentracker aus Volkhardt u. a. (2013a). Abbildung 4.1 gibt einen konzeptionellen Überblick über den Ablauf des Trackings. Alle Hypothesen der Detektoren liegen nach den Verarbeitungsschritten aus Abschnitt 3.5 in Form von 3D Gaußverteilungen in Weltkoordinaten vor. Zusätzlich besitzt jede Detektion eine Existenzwahrscheinlichkeit (Abschnitt 4.8). Das Tracken in Weltkoordinaten birgt den Vorteil, dass auf eine Eigenbewegungskompensation verzichtet werden kann. Für die Eigenbewegungskompensation wird in der Regel ein eigener Schätzer benötigt, welcher in jedem Schritt die Bewegung des Roboters

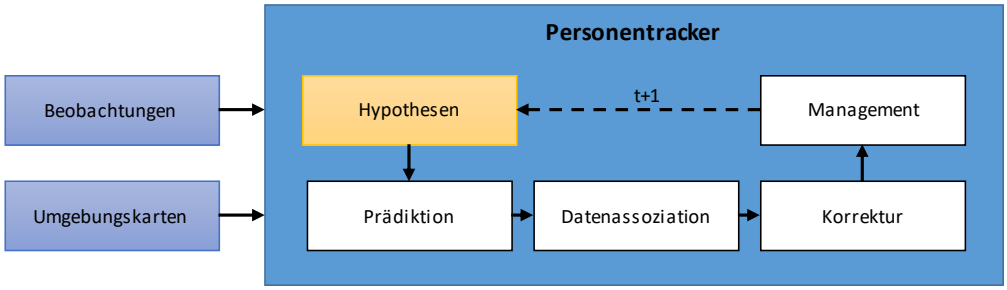


Abbildung 4.1.: Konzeptioneller Aufbau des Personentrackers. Nachdem die Hypothesen in den neuen Zeitschritt prädiziert wurden, erfolgt die Datenassoziation der Beobachtungen und der Korrekturschritt. Anschließend regelt eine Managementeinheit die Zahl der Hypothesen. Als optionales Wissen geht die Umgebungskarte ein.

schätzt und diese mit der zugehörigen Unsicherheit aus den Hypothesen herausrechnet (Schubert 2011; Richter 2012).
Zunächst werden die vorhandenen Hypothesen im Personentracker nach dem Bayes-Filter Prinzip prädiziert (Abschnitt 4.4). Anschließend erfolgt die Datenassoziation, welche neue Beobachtungen zu vorhandenen Hypothesen zuordnet oder neue Personentracks erzeugt. Unabhängige Hypothesen von unterschiedlichen Detektoren werden mittels Kalman-Filterung und abhängige Detektionen mittels Covariance Intersection fusioniert. Als Gating Bereich für die Zuordnung wird die Mahalanobis Distanz verwendet. Da das System mit asynchronen Detektionsmodulen arbeitet, können Beobachtungen verspätet (engl. *out-of-sequence*) im System ankommen und müssen berücksichtigt werden (Abschnitt 4.7.2). Die Korrektur der Hypothesen mit den Beobachtungen erfolgt mittels Bayes-Filterung (Abschnitt 4.4). Als Besonderheit dieser Arbeit erlaubt die Architektur des Personentrackers verschiedene Bayes-Filter, z. B. Extended- und Unscented Kalman-Filter, zu verwenden und deren Systemmodelle auszutauschen (Abschnitte 4.5 und 4.6). Für jede Hypothese schätzt der Personentracker zusätzlich eine Existenzwahrscheinlichkeit (Abschnitt 4.8). Zuletzt existiert eine Management-Komponente, die die Entstehung, Löschung und Plausibilitätsprüfung von Hypothesen übernimmt (Abschnitt 4.9).

4.4. Bayes-Filterung

In dieser Arbeit wird der Zustand einer Person mittels Bayes-Filter Algorithmen geschätzt. Bei einem Bayes-Filter handelt es sich um einen rekursiven Zustandsschätzer, der den Systemzustand als Wahrscheinlichkeitsverteilung auf-

fasst. Der Zustand x_t kann dabei nicht beobachtbare Größen enthalten, die durch den Filter geschätzt werden. Hierfür stellt der Filter über Systemmodelle eine Verbindung zwischen der Zustandsschätzung $bel(x_t)$, den Messdaten z_t und der Steuerung u_t her. Die Systemmodelle werden in der Prädiktion und dem Beobachtungsupdate verwendet (Abschnitt 4.5).
Der prinzipielle Ablauf für alle Bayes-Filtervarianten ist in Algorithmus 4.1 dargestellt.

Algorithmus 4.1 : Bayes-Filter (vgl. Thrun u. a. (2005))

Eingabe : $bel(x_{t-1}), u_t, z_t$ // Zustandsschätzung, Steuerung, Beobachtung

1 **foreach** x_t **do** // Für alle möglichen Zustände x_t
2 $\quad \overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1}$ // Prädiktion
3 $\quad bel(x_t) = \eta\, p(z_t \mid x_t) \overline{bel}(x_t)$ // Beobachtungsupdate

Ausgabe : $bel(x_t)$

Die aktuelle Zustandsschätzung $bel(x_t) = p(x_t \mid z_{1:t}, u_{1:t})$ ergibt sich rekursiv aus der vorherigen Schätzung $bel(x_{t-1})$ und der aktuellen Beobachtung z_t. Der Steuerungsterm u_t kann beim Personentracking ignoriert werden, da das System keinen Einfluss auf das Verhalten der Personenzustände nehmen kann. Für alle möglichen Zustände x_t wird zunächst der vorläufige Systemzustand $\overline{bel}(x_t) = p(x_t \mid z_{1:t-1}, u_{1:t})$ mittels Prädiktion berechnet. Anschließend erfolgt die Korrektur mit der aktuellen Beobachtung im Update-Schritt. Der Bayes-Filter beschreibt die mathematischen Modelle als Verteilungsdichtefunktionen, schränkt die Umsetzung der Funktionen jedoch nicht ein. Je nach Zustands- und Modellrepräsentation spricht man von unterschiedlichen Filterarten (Thrun u. a. 2005). Die folgenden Abschnitte beschreiben die Filter, welche in dieser Arbeit verwendet werden. Eine ausführlichere Erläuterung zur probabilistischen Zustandsschätzung findet sich Thrun u. a. (2005) und Murphy (2012).

4.4.1. Kalman-Filter

Der Kalman-Filter repräsentiert einen kontinuierlichen Systemzustand eines Bayes-Filters als Gaußverteilung (Kalman 1960; Barker u. a. 1994). Er entspricht einer optimalen Filterung, falls der Systemzustand tatsächlich normalverteilt ist und Prädiktion sowie Beobachtungsupdate lineare Funktionen mit additivem gaußschem Rauschen sind.
Abbildung 4.2 zeigt das graphische Modell der Kalman-Filterung. In jedem Zeitschritt wird der Zustandsvektor $\mathbf{x}_{t-1}$ zur neuen Zustandsschätzung $\mathbf{x}_t$ pro-

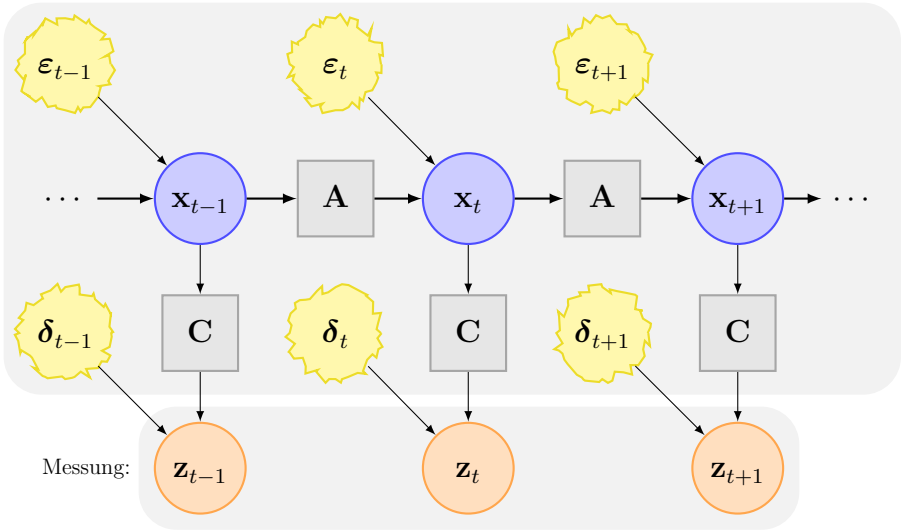


Abbildung 4.2.: Kalman-Filter. Der Zustand $\mathbf{x}$ wird über Matrix $\mathbf{A}$ propagiert. Die Korrektur mit Messung $\mathbf{z}$ erfolgt über die Beobachtungsmatrix $\mathbf{C}$. Weiterhin bezeichnen $\boldsymbol{\varepsilon}$ und $\boldsymbol{\delta}$ das System- und Messrauschen[a].

[a] Vgl. Lingner-2010: `http://www.texample.net/tikz/examples/kalman-filter`

pagiert, indem die konstante Transitionsmatrix $\mathbf{A}_t$ angewendet wird und der Rauschterm $\boldsymbol{\varepsilon}_t$ addiert wird:

$$\mathbf{x_t} = \mathbf{A}_t\mathbf{x}_{t-1} + \mathbf{B}_t\mathbf{u}_t + \boldsymbol{\varepsilon}_t \tag{4.1}$$

Matrix $\mathbf{B}_t$ stellt einen Zusammenhang zwischen der Steuerung $\mathbf{u}_t$ und dem Zustand $\mathbf{x_t}$ her. Der externe Steuerungsterm $\mathbf{u}_t$ wird dabei im Folgenden und in Abbildung 4.2 aufgrund fehlender Relevanz für das Personentracking ignoriert. Das Prozessrauschen $\boldsymbol{\varepsilon}_t$ wird als mittelwertfreies additives weißes gaußsches Rauschen mit Kovarianz:

$$\mathbb{E}[\boldsymbol{\varepsilon}_t\boldsymbol{\varepsilon}_t^T] = \mathbf{Q}_t \tag{4.2}$$

angenommen.

Der Zustandsvektor $\mathbf{x}_t$ wird anschließend durch die Beobachtung korrigiert. Der Beobachtungsvektor $\mathbf{z}_t$ ergibt sich dabei durch die Multiplikation aus aktuellem Zustand $\mathbf{x}_t$ und der Beobachtungsmatrix $\mathbf{C}_t$ sowie Messrauschen $\boldsymbol{\delta}_t$:

$$\mathbf{z}_t = \mathbf{C}_t\mathbf{x}_t + \boldsymbol{\delta}_t \tag{4.3}$$

Das Messrauschen $\boldsymbol{\delta}_t$ wird ebenfalls als mittelwertfreies additives gaußsches Rauschen mit Kovarianz:

$$\mathbb{E}[\boldsymbol{\delta}_t \boldsymbol{\delta}_t^T] = \mathbf{R}_t \tag{4.4}$$

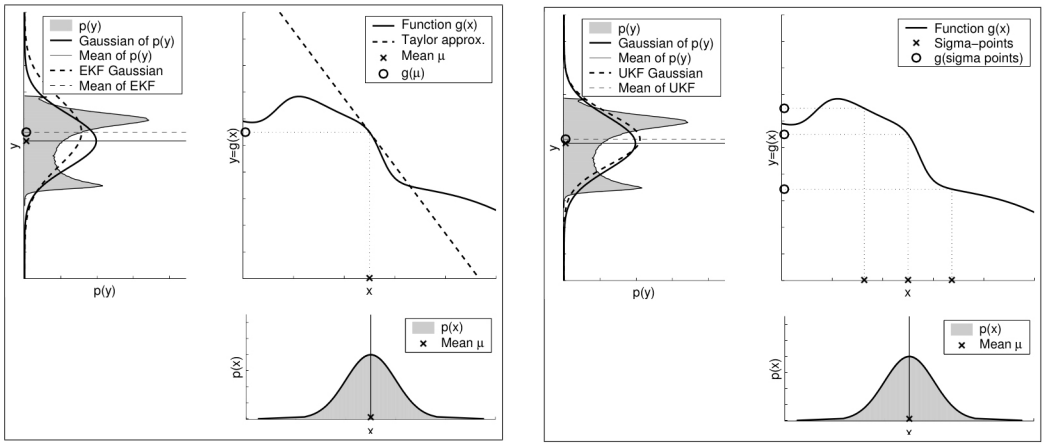
modelliert. Die algorithmischen Details des Kalman-Filters mit Prädiktion und Korrekturschritt sind in Anhang A.5.1 zu finden.

4.4.2. Extended Kalman-Filter

Der Extended Kalman-Filter (EKF) erlaubt nichtlineare Prädiktions- und Beobachtungsfunktionen (Welch u. a. 1995) indem er die Modellfunktionen durch eine Taylorreihe linearisiert (Thrun u. a. 2005, Seite 56 ff.). Die Linearisierung erfolgt dabei im Prädiktionsschritt ausgehend vom Erwartungswert der letzten Schätzung, also dem Mittelwert $\boldsymbol{\mu}_{t-1}$ der Gaußverteilung. Analog wird beim Beobachtungsupdate der neu geschätzte Systemzustand $\bar{\boldsymbol{\mu}}_t$ als Linearisierungspunkt verwendet. Bei mehrdimensionalen Systemen wird die Taylorreihe mittels Jacobimatrizen realisiert. Die mathematischen und algorithmischen Details des Extended Kalman-Filters finden sich in Anhang A.5.2. Zur Veranschaulichung ist die Linearisierung beim Prädiktionsschritt in Abbildung 4.3(a) bildlich dargestellt. In dieser Arbeit wird die dargestellte nichtlineare Funktion $g(x)$ für ein nichtlineares Systemmodell mit konstanter Orientierung und Geschwindigkeit verwendet (siehe Abschnitt 4.5). Die spezielle Implementierung von $g(x)$ für den EKF ist in Anhang A.6.2 gegeben. Der Vorteil der Linearisierung ist die schnelle Berechenbarkeit. Nachteilig wirkt sich der Linearisierungsfehler aus, welcher je nach lokaler Nichtlinearität der Funktionen und Varianz des Systemzustands unterschiedlich groß ist.

4.4.3. Unscented Kalman-Filter

Der Unscented Kalman-Filter (UKF) wurde von S. J. Julier u. a. (1997) vorgestellt. Die zugrunde liegende Unscented Transformation beruht auf der Idee, dass es im Allgemeinen einfacher ist, eine Gaußverteilung zu approximieren als eine (nichtlineare) Funktion (S. Julier u. a. 1995). Im Gegensatz zum EKF wird die nichtlineare Funktion auf einem deterministisch gewählten Set an Punkten, benannt als Sigma Punkte, angewendet, und anschließend eine Gaußverteilung aus den transformierten Punkten geschätzt (Murphy 2012). Die Unscented Transformation erfasst und transformiert daher Mittelwert und Kovarianz mittels der gewichteten Sigma Punkte, welche anhand der Gaußverteilung berechnet werden. Da die Sigma Punkte im Gegensatz zum Partikel-Filter nicht zufällig gezogen werden, kann keine beliebige Funktion approximiert werden. Es werden jedoch auch sehr viel weniger Punkte zur Approximation benötigt (Bellotto u. a. 2009).



(a) Extended Kalman-Filter **(b)** Unscented Kalman-Filter

Abbildung 4.3.: Vergleich von Extended- und Unscented Kalman-Filter. Die Grafiken zeigen eine nichtlineare Funktion $g(x)$ (Mitte schwarz), welche den normalverteilten Zustand $p(x)$ (unten grau) in einen neuen bimodalen Zustand $p(y)$ überführt (links grau). Die exakte Gaußapproximation von $p(y)$ ist jeweils links in schwarz dargestellt. (a) Die nichtlineare Funktion wird beim EKF an der Stelle μ des Zustands $p(x)$ linearisiert (Mitte gestrichelte Linie), um die normalverteilte Approximation der Verteilung $p(y)$ zu erhalten (links gestrichelte Verteilung). (b) In der Abbildung verwendet der Unscented Kalman-Filter drei Sigma Punkte, um die nichtlinearer Funktion $g(x)$ zu approximieren. Aus den transformierten Sigma Punkten wird die normalverteilte Approximation der Verteilung $p(y)$ bestimmt. Im Vergleich zu (a) approximiert der UKF die tatsächliche Normalverteilung besser als der EKF. Quelle: Thrun u. a. (2005)

Die prinzipielle Funktionsweise des UKFs bei der Prädiktion des Systemzustands mit einem nichtlinearen Systemmodell (siehe Abschnitt 4.5) ist in Abbildung 4.3(b) dargestellt. Im Vergleich zum EKF fällt der Fehler zur exakten Gaußverteilung geringer aus. Die mathematischen und algorithmischen Details des Unscented Kalman-Filters sind in Anhang A.5.3 zu finden. Der UKF ist im Vergleich zum EKF berechnungsintensiver,liefert aber eine höhere Genauigkeit, da der UKF bis zum zweiten Taylorglied, der EKF aber nur bis zum ersten Taylorglied, eine exakte Lösung berechnet. Dadurch verbessert sich die Schätzung vor allem bei Systemzuständen mit hoher Varianz. Als weiterer Vorteil werden keine Ableitungen der Systemmodelle benötigt (S. Julier u. a. 1995; S. J. Julier u. a. 2000).

4.4.4. Partikel-Filter

Der Partikel-Filter approximiert eine beliebige Wahrscheinlichkeitsverteilung durch eine begrenzte Anzahl von gewichteten Partikeln, welche jeweils einen diskreten Systemzustand repräsentieren (Isard u. a. 1998; Rubinstein u. a. 2008). Jedes Partikel durchläuft die Prädiktion und wird anschließend anhand der Beobachtung im Beobachtungsupdate neu gewichtet. Durch eine zufällige gewichtete Auswahl von Partikeln werden nicht zur Beobachtung passende Partikel ausgedünnt und passende verstärkt (engl. *survival of the fittest*). Die Verteilung der Partikel repräsentiert anschließend approximativ die Wahrscheinlichkeitsverteilung des Systemzustands. Partikel-Filter haben den Vorteil der einfachen Implementierbarkeit. Als nachteilig erweist sich die mangelnde Effizienz in hochdimensionalen Zustandsräumen[1]. Weiterhin können degenerierte Anfangszustände zu Problemen bei der Verteilung der Partikel führen (Arulampalam u. a. 2002). Eine ausführliche mathematische und algorithmische Abhandlung des Partikel-Filters ist in (Arnaud Doucet 2001; Thrun u. a. 2005; Murphy 2012) zu finden. Im entwickelten Personentracking-Frameworks ist es möglich, Partikel-Filter als Zustandsschätzer zu verwenden. Allerdings werden diese aufgrund des hohen Berechnungsaufwands nicht im Anwendungsszenario eingesetzt und evaluiert.

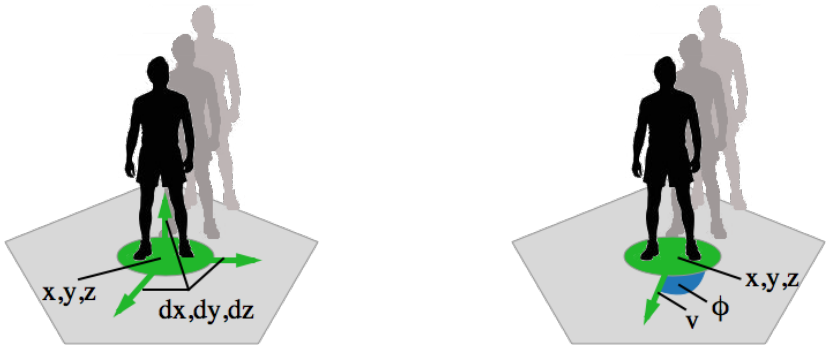
4.5. Systemmodelle

Systemmodelle beschreiben die Eigenschaften und das Verhalten eines Systems. Sie unterscheiden sich je nach Szenario und bilden die mathematische Grundlage für jede Implementierung eines Bayes-Filters. Speziell spezifiziert ein Systemmodell für einen Anwendungsfall:

1. Systemzustand
2. Bewegungsmodell
3. Beobachtungsmodell

Der Systemzustand $\mathbf{x}_t$ beschreibt die zu schätzenden Variablen des Systems. Das Bewegungsmodell stellt eine Realisierung der Wahrscheinlichkeitsverteilung $p(x_t \mid u_t, x_{t-1})$ des Bayes-Filters (Algorithmus 4.1) dar. Das Beobachtungsmodell realisiert die Wahrscheinlichkeitsverteilung $p(z_t \mid x_t)$ des Bayes-Filters. Je nach Szenario und eingesetztem Bayes-Filter fällt die Implementierung des Systemmodells leicht unterschiedlich aus. Beispielsweise kann der

[1] Die Anzahl der benötigten Partikel wächst exponentiell mit der Anzahl der Zustandsdimensionen.



(a) Konstante Geschwindigkeit

(b) Konstante Orientierung

Abbildung 4.4.: Lineares Systemmodell mit konstanter Geschwindigkeit und nichtlineares Systemmodell mit konstanter Orientierung. Quelle: Arenknecht (2015)

Systemzustand als Gaußverteilung oder als Partikelverteilung und das Bewegungsmodell linear oder nichtlinear implementiert werden.
Im Rahmen dieser Arbeit, speziell in Volkhardt u. a. (2013a) und Arenknecht (2015)[2], werden zwei Systemmodelle untersucht, die die Bewegung von Personen im Raum modellieren.

1. Konstante Geschwindigkeit (Volkhardt u. a. 2013a)
2. Konstante Orientierung und Geschwindigkeit (Bellotto u. a. 2009)

Im ersten Fall handelt es sich um ein lineares und im zweiten Fall um ein nichtlineares Systemmodell. In Abbildung 4.4 ist der Zustand beider Modelle grafisch dargestellt. Modell 1 (Abbildung 4.4(a)) nach Volkhardt u. a. (2013a) repräsentiert den Zustand einer Person mit:

$$\mathbf{x}_t = (x_t, y_t, z_t, \dot{x}_t, \dot{y}_t, \dot{z}_t) \tag{4.5}$$

Der Zustand einer Person wird über deren Position im Raum und deren Geschwindigkeit in jede Raumrichtung beschrieben. Als lineares Bewegungsmodell wird ein konstantes Geschwindigkeitsmodell mit zufälliger Beschleunigung, die als gaußsches Rauschen modelliert wird, verwendet. Die mathematische Beschreibung des Modells findet sich in Anhang A.6.1
Modell 2 (Abbildung 4.4(b)) aus Bellotto u. a. (2009) repräsentiert den Zustand einer Person mit:

$$\mathbf{x}_t = (x_t, y_t, z_t, \phi_t, v_t) \tag{4.6}$$

[2]Vom Autor im Rahmen dieser Arbeit betreut.

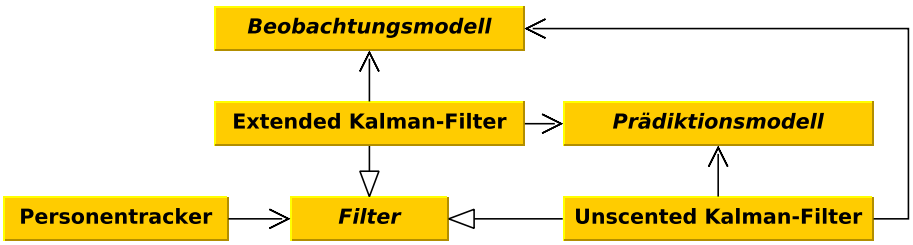


Abbildung 4.5.: UML-Klassendiagramm. Der Personentracker besitzt als Property eine abstrakte Filterklasse, die durch verschiedene Kalman-Filter implementiert werden kann. Jeder Filter besitzt eine abstrakte Prädiktions- und Beobachtungsklasse, welche durch die jeweiligen Systemmodelle implementiert werden. Die Spezialisierungen der jeweiligen Systemmodelle sind aufgrund der Übersichtlichkeit nicht dargestellt.

Der Zustand einer Person definiert sich demnach über der Position, der Orientierung in der x-y Ebene und der Geschwindigkeit entlang der Orientierung. Die mathematische Beschreibung des Systemmodells findet sich in Anhang A.6.2.
Als Beobachtung wird bei beiden Modellen die Position einer Person im Raum verwendet:

$$\mathbf{z}_t = (x'_t, y'_t, z'_t) \tag{4.7}$$

4.6. Softwaretechnische Umsetzung der Filter und Systemmodelle

Um während der Entwicklung des Roboterassistenten verschiedene Filter und Systemmodelle evaluieren zu können, wurde der Personentracker so umgesetzt, dass Filter und Systemmodelle austauschbar sind, indem diese jeweils abstrakte Filter- bzw. Systemmodellklassen implementieren. Das bedeutet, dass prinzipiell jeder Filter genutzt werden kann, solange er mit 3D-Hypothesen (Gaußverteilungen) als Eingabe umgehen kann und seinen geschätzten Zustand als Gaußverteilung ausgeben kann. Daher leiten alle vorgestellten Filterverfahren aus Abschnitt 4.4 von einer gemeinsamen Basisklasse ab und können per Konfigurationsdatei gesetzt werden. Das entsprechende UML-Diagramm ist in Abbildung 4.5 dargestellt.
In der Realisierung implementieren alle Filter folgende Methoden:

- Initialisierung: Der Startzustand und die Startkovarianz werden festgelegt.

- Prädiktion: Der Zustand wird über eine Referenz zu einem Bewegungsmodell in den neuen Zeitschritt prädiziert.
- Update: Der interne Zustand wird mittels Referenz auf ein Beobachtungsmodell mit den aktuellen Beobachtungen korrigiert.
- Output: Der interne Filterzustand wird in eine Normalverteilung umgewandelt.

Die zugehörigen Systemmodelle leiten ebenfalls von einer gemeinsamen Basisklasse ab und sind per Konfigurationsdatei austauschbar. Dies erlaubt es in der Entwicklungs- und Evaluationsphase schnell zwischen verschiedenen Modellen umzuschalten (Abschnitt 4.5). Weiterhin können die Modelle bei Passfähigkeit in anderen Filtern wiederverwendet werden. Beispielsweise lässt sich ein Bewegungsmodell eines linearen Kalman-Filters auch im EKF nutzen.
Ferner erlaubt die Umsetzung, dass sich mehrere Filter denselben Systemzustand teilen können. Dies ermöglicht es für verschiedene Situationen, beispielsweise ein stehender und sitzender Nutzer, angepasste Filter und Systemmodelle zu verwenden. Einen ähnlichen Weg beschreiten *Interacting Multi-Models*, welche mehrere Prädiktionsmodelle gleichzeitig nutzen und anschließend das Modell wählen, dessen Vorhersagen am besten zu den aktuellen Beobachtungen passen (Spinello u. a. 2010a; Madrigal u. a. 2013).

4.7. Datenassoziation

Unter Datenassoziation wird die Zuordnung der Beobachtungen zu den Hypothesen im Personentracker verstanden. Alle Beobachtungen der unterschiedlichen Detektoren seit dem letzten Updateschritt werden nach ihrer Aufnahmezeit sortiert und sequenziell verarbeitet.
Zunächst werden alle Hypothesen auf den Zeitpunkt der aktuellen Beobachtung prädiziert. Anschließend wird die Beobachtung b der Hypothese h mit der geringsten Mahalanobis-Distanz[3] zugeordnet (Uhlmann 2001):

$$d(h,b) = (\boldsymbol{\mu}_h - \boldsymbol{\mu}_b)^T (\boldsymbol{\Sigma}_h + \boldsymbol{\Sigma}_b)^{-1} (\boldsymbol{\mu}_h - \boldsymbol{\mu}_b) \;, \tag{4.8}$$

wobei $\boldsymbol{\mu}_h = (x_h, y_h, z_h)$ und $\boldsymbol{\mu}_b = (x_b, y_b, z_b)$ den Mittelwert des Positionsanteils der Tracker-Hypothese beziehungsweise der Beobachtung bezeichnen. $\boldsymbol{\Sigma}_h$ und $\boldsymbol{\Sigma}_b$ repräsentieren die entsprechenden Kovarianzen. Die Datenassoziation

[3] Die Mahalanobis-Distanz ist die mehrdimensionale Verallgemeinerung der Distanz eines Punkts p vom Mittelwert einer Datenmenge D gemessen in Standardabweichungen.

erfolgt im Zustandsraums und nicht im Beobachtungsraum[4]. Das bedeutet, dass Beobachtungen in Weltkoordinaten auf Basis ihrer Mahalanobis-Distanz zu den Hypothesen zugeordnet werden. Nach erfolgreicher Zuordnung erfolgt der Korrekturschritt des Bayes-Filters.
Zur Generierung neuer Hypothesen wird ein Gating eingesetzt (Bar-Shalom u. a. 2002). Dabei wird eine Detektion b als neuer Track angesehen, wenn die Mahalanobis-Distanz zu allen Hypothesen h_i über einer Schwelle:

$$d(h_i, b) \geq \gamma \quad \forall\, h_i \tag{4.9}$$

liegt, also davon ausgegangen werden kann, dass die Beobachtung nicht von einer vorhandenen Hypothese generiert wurde. Der Wert γ wird als Gate-Schwellwert bezeichnet und bestimmt die Wahrscheinlichkeit, dass sich die Beobachtung innerhalb der, durch die Mahalanobis Distanz aufgespannten, Region befindet (Bar-Shalom u. a. 1996, S. 95 ff.).
Mit der Feststellung, dass die quadrierten Distanzen $d^2(h, b_j)$ der Beobachtungen (Samples) einer Normalverteilung Chi-Quadrat verteilt mit n Freiheitsgraden sind, $d^2(h, b) \sim \chi^2_n$, kann ein passender Schwellwert über die Quantilfunktion der Chi-Quadrat-Verteilung bestimmt werden:

$$\gamma = \sqrt{chi2_{inv}(p, n)}\ . \tag{4.10}$$

Hier bezeichnet $chi2_{inv}$ die inverse Qui-Quadrat-Verteilungsfunktion und $p = 0.1$ bestimmt sich aus dem 90% Quantil. Die Freiheitsgrade werden auf $n = 3$ gesetzt. Mit den gegebenen Parametern ergibt sich $\gamma = 2.5$.
In anderen Ansätzen wird dieses Gating auch verwendet, um zu entscheiden, welche Hypothesen als Quelle für bestimmte Beobachtungen in Frage kommen (Bellotto u. a. 2007). Damit kann der Rechenaufwand beim Transformieren des Zustandes in den Beobachtungsraum verringert werden oder Hypothesen im *Multi-Hypotheses-Tracking* ausgeschlossen werden (Bar-Shalom u. a. 1996; Reid 1979). Zur Generierung der Hypothese wird die Position aus der Beobachtung übernommen, die Geschwindigkeit des Zustandes $\mathbf{x}_k$ mit 0 und die Kovarianz mit 1 initialisiert.
Als Alternative wurde der Kuhn-Munkres Algorithmus zur optimalen Zuordnung aller Beobachtungen zu den Hypothesen mit geringstem globalen Abstand untersucht (Kuhn u. a. 1955). Dieser lieferte keine Verbesserungen der Evaluationsmetriken im untersuchten Szenario (Abschnitt 4.10). Weitere Untersuchungen zu den verschiedenen Zuordnungsvariationen finden sich in Uhlmann (2001) und Konstantinova u. a. (2003).

[4]Die Hypothesen werden nicht, wie üblich, in den Beobachtungsraum transformiert, da die Beobachtungen bereits in Weltkoordinaten vorliegen.

Einen weiterführenden Ansatz verfolgen *Multi-Hypotheses-Tracking* Ansätze, welche mehrere mögliche Assoziationen und Hypothesen berücksichtigen (Reid 1979; Cox 1993; Kim u. a. 2015). Einen probabilistischen Ansatz bei der Datenassoziation verfolgen *Integrated-* bzw. *Joint Probabilistic Data Association Filter* (Musicki u. a. 1994; Fortmann u. a. 1983). Diese Verfahren sind aber nicht Gegenstand der Arbeit.

4.7.1. Covariance Intersection

Ein Beobachtungsmodul liefert unter Umständen innerhalb eines Zeitschrittes mehrere Beobachtungen an nahegelegenen Positionen[5], welche während der Datenassoziation mit einer Hypothese fusioniert werden würden. Unter der Annahme, dass diese Beobachtungen nur von einer Quelle stammen, ist die Korrelation zwischen den Beobachtungen im Allgemeinen unbekannt. In diesem Fall würde eine Fusion im Bayes-Filter, welcher Unabhängigkeit der Beobachtungen voraussetzt, die Kovarianz der Beobachtung unterschätzen. Daher wird in diesem Fall die *Covariance Intersection* verwendet, um alle Beobachtungen zu einer Gaußverteilung zu fusionieren (S. Julier u. a. 1997; S. J. Julier u. a. 2001; Chen u. a. 2002):

$$\boldsymbol{\Sigma}_3^{-1} = (1-\omega)\boldsymbol{\Sigma}_1^{-1} + \omega\boldsymbol{\Sigma}_2^{-1} \,, \tag{4.11}$$

wobei ω einen Wichtungsparameter bezeichnet, der den Einfluss der Beobachtungskovarianzen $\boldsymbol{\Sigma_1}$ und $\boldsymbol{\Sigma_2}$ auf die resultierende Kovarianz $\boldsymbol{\Sigma_3}$ kontrolliert. Er bestimmt sich mit:

$$\omega = \frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_1|+|\boldsymbol{\Sigma}_2|} \,. \tag{4.12}$$

Der Mittelwert der fusionierten Beobachtungen ist wie folgt bestimmt:

$$\boldsymbol{\mu}_3 = \boldsymbol{\Sigma}_3 \left[(1-\omega)\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \omega\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\right] \,. \tag{4.13}$$

4.7.2. Beobachtungen außer der Reihe

Bei der Nutzung asynchroner Sensoren mit unterschiedlicher Laufzeit kommt es zwangsläufig zu Beobachtungen außer der Reihe (engl. *out of sequence measurements*). Benötigen die Detektionsmodule unterschiedlich lange für die Verarbeitung ihrer jeweiligen Sensordaten, kommt es häufiger vor, dass die Detektion eines Detektors mit hoher Latenz erst dann im Personentracker ankommt,

[5] Dies ist häufig bei Sliding-Window Detektoren der Fall, wenn der Klassifikator in vielen, nur um wenige Pixel verschobenen, Detektionsfenstern eine Person erkennt und keine *Non-Maximum Suppression* erfolgt.

nachdem bereits eine aktuellere Detektion eines Detektors mit geringer Latenz verarbeitet wurde (Kaempchen u. a. 2003). Die Arbeit präsentiert drei Lösungsansätze für dieses Problem (Groves 2013).

Erneutes Filtern

Bei diesem Ansatz werden alle Beobachtungen und die zugehörigen geschätzten Zustände in einer Liste gespeichert. Sobald eine Messung mit veraltetem Zeitstempel beim Personentracker eintrifft, wird diese an die richtige Stelle in der Liste eingefügt und mit allen nachfolgenden Beobachtungen verarbeitet. Dies führt zum erneuten Anwenden der Bayes-Filter Gleichungen, um erneut zum aktuellen Zeitpunkt aufzuholen. Als weiterer Nachteil müssen alle Beobachtungen und geschätzten Zustände für mindestens den Zeitraum der größten Latenz eines Sensors gespeichert werden.

Nichtdeterministisches Puffern

Ziel des nichtdeterministischen Pufferns (engl. *non-deterministic buffering*) ist die Umsortierung der Beobachtungen nach dem Zeitstempel ihrer Aufnahme (Kaempchen u. a. 2003). Die Idee ist die Speicherung aller Beobachtungen, bis die nächste veraltete Beobachtung des Sensors mit der höchsten Latenz ankommt. Dazu besitzt der Personentracker für jeden Detektor eine Warteschlange, welche die Beobachtungen mit dem Messzeitpunkt des jeweiligen Sensors aufsammelt. Im Gegensatz zum deterministischen Puffern ist die maximale Latenz der Sensoren in diesem Fall unbekannt und nicht konstant. Durch Verwendung eines Triggers zu festgelegten Zeitpunkten werden die Warteschlangen abgearbeitet. Dabei werden so lange Beobachtungen sequenziell aus den einzelnen Puffern verarbeitet, bis sich in einer Warteschlange eines Sensors keine Beobachtungen mehr befinden. Daraufhin wird die Verarbeitung unterbrochen und auf den nächsten Trigger gewartet.
Durch das Puffern entspricht der aktuelle interne Zeitstempel des Personentrackers stets einem Zeitpunkt in der Vergangenheit. Die maximale Latenz beträgt dabei:

$$L_{max} = \max(L_{Sensor}) + \max(T_{Sensor}) \tag{4.14}$$

wobei L_i die Latenz und T_i die Zykluszeit des Sensors i angeben.
Werden die geschätzten Zustände des aktuellen Zeitpunktes benötigt, kann der Prädiktionsschritt des Personentrackers verwendet werden, um die Hypothesen (ohne die Information neuerer Beobachtungen) in der Zeit vorwärts zu schätzen. Als Nachteil kann der Ansatz nicht mit einem Sensorausfall umgehen, sodass die Asynchronizität und Modularität des Systems eingeschränkt

wird und in der Praxis ein Monitoring für Sensorausfälle implementiert werden muss.

Rückwärtsprädiktion

Die Rückwärtsprädiktion (engl. *Retrodiction*) prädiziert den aktuellen Zustand des Systems in umgekehrter Richtung (mit negativem Δt) auf den Zeitpunkt der veralteten Beobachtung. Anschließend kann diese verarbeitet werden und das System wieder auf den aktuellen Zustand prädiziert werden. Wie in Bar-Shalom (2002) gezeigt, verletzt die Rückwärtsprädiktion allerdings die Annahme des Bayes-Filters, dass das Prozessrauschen unabhängig vom Systemzustand ist. Daher stellt Bar-Shalom (2002) mehrere Methoden vor, bei denen zwei approximative Methoden das Prozessrauschen bei der Rückwärtsprädiktion ignorieren ($\boldsymbol{\varepsilon}_t = 0$). Die vorgestellte exakte Methode modelliert das Prozessrauschen während der Retrodiction in einem relativ komplexen Verfahren. Eine weitere Methode findet sich in Rheaume u. a. (2008), bei der eine Dekorrelationsmethode genutzt wird.
In der Evaluation (Abschnitt 4.10) kommt die approximative Methode der Retrodiction, bei der das Prozessrauschen während der Rückwärtsprädiktion ignoriert wird, zum Einsatz.

4.8. Schätzung der Existenzwahrscheinlichkeit

Die Existenzwahrscheinlichkeit kann genutzt werden, um sich zwischen mehreren Hypothesen auf einen Interaktionspartner zu konzentrieren oder als Basis dienen, um Hypothesen aus der Trackliste zu entfernen (Abschnitt 4.9). Sie gibt an, wie sicher es sich bei dem getrackten Objekt um eine Person handelt. Neben dem bisher vorgestellten Bayes'schen Zustandstracking modelliert diese Arbeit für jede Hypothese eine zusätzliche Existenzwahrscheinlichkeit. Zur Schätzung der Existenzwahrscheinlichkeit $p(H)$ wird ein Naïve Bayes Klassifikator verwendet. Der Klassifikator schätzt die Existenzwahrscheinlichkeiten $p(H_k)$ mit $H = \{Person, Objekt\}$ mithilfe der einzelnen Features $x_1, \ldots, x_n$. Die Features entsprechen den Detektionsmodulen, die die Hypothese beobachtet haben. Unter der Annahme, dass die Features nur von der Klasse C abhängen und bei gegebener Klasse untereinander unabhängig sind[6], ergibt sich die Formel:

$$p(H_k \mid x_1, \ldots, x_n) = \frac{1}{Z}\, p(H_k) \prod_{i=1}^{n} p(x_i \mid H_k)\ , \tag{4.15}$$

[6] Bedingte Unabhängigkeit.

wobei $\frac{1}{Z}$ einem Normalisierungsfaktor entspricht. Die jeweiligen Wahrscheinlichkeiten $p(x_i \mid C_k)$ für einen Sensor ergeben sich aus der Richtig-positiv-Rate (TPR) für die Klasse Person, beziehungsweise der Falsch-positiv-Rate (FPR) für die Klasse Objekt. Die Raten werden für jeden Detektor auf repräsentativen Testdatensätzen gewonnen (Abschnitt 4.10.2). Der Klassenprior $p(C_k)$ kann jeweils mit 0.5 angenommen werden, oder aus einem gelabelten Trainingsdatensatz geschätzt werden. Intuitiv steigt die Existenzwahrscheinlichkeit, wenn die Hypothese von mehreren Detektoren mit hoher TPR und geringer FPR beobachtet wird. Wird die Hypothese von einem Detektor mit niedriger TPR und hoher FPR beobachtet, ist die Existenzwahrscheinlichkeit entsprechend geringer.
Weitere Möglichkeiten zur Schätzung der Existenzwahrscheinlichkeit umfassen den Einsatz von Bayes-Filtern, IPDA-Filter oder die Dempster-Shafer Theorie (Altendorfer u. a. 2010; Maehlisch u. a. 2007; Gehrig u. a. 2012; Aeberhard u. a. 2011). Im Gegensatz zur vorgestellten Lösung müssen bei diesen Verfahren die Sensorsichtbereiche und Detektionswahrscheinlichkeiten exakt modelliert werden. Dabei müssen alle Sensoren das Objekt in ihrem Sichtbereich mit der modellierten Wahrscheinlichkeit erkennen. Bei ungenauer Modellierung reicht bereits ein einzelner Sensor mit genügend Beobachtungen aus, um die Existenzwahrscheinlichkeit nahe dem Maximum zu bringen[7].
Im vorliegenden Anwendungsfall stellt sich, anders als im Automobilbereich, eine exakte Modellierung der Detektionsmodelle als schwierig heraus. Auch wenn sich die Person im Sensorsichtbereich befindet, müssen das Gesicht oder die Beine nicht immer detektierbar sein. Dies tritt beispielsweise auf, wenn die Person vom Roboter abgewandt ist oder auf einem Sofa sitzt und die Beine verdeckt sind. Weiterhin ist im häuslichen Bereich mit den eingesetzten Verfahren und Sensoren mit geringerer TPR und höherer FPR zu rechnen[8].

4.9. Trackmanagement und Umgebungswissen

Neben dem eigentlichen Tracking der Hypothesen besitzt der Personentracker eine Logik zur Generierung von neuen Tracks und dem Löschen von vorhandenen Tracks (Abschnitt 4.9.1). Ein Track stellt dabei eine kontinuierlich getrackte Hypothese mit einer eindeutigen ID dar. Zusätzlich kann Umgebungswissen einfließen, um die Generierung und Beendigung von Tracks zu beeinflussen (Abschnitt 4.9.2).

[7] In der Praxis behilft man sich meist mit einem Prozessmodell welches die Existenzwahrscheinlichkeit über die Zeit abklingen lässt.

[8] Bar-Shalom u. a. (1996) geben bei einer typischen Radaranwendung eine Detektionswahrscheinlichkeit von 0.95 und eine Falsch-Alarm-Rate von 10^{-8} pro Auflösungszelle an.

4.9.1. Generierung und Löschung von Tracks

Neue Tracks werden mit jeder Beobachtung, welche sich nicht zu einer vorhandenen Hypothese zuordnen lässt, generiert.
Um die Anzahl der Hypothesen im Tracker zu limitieren, werden Tracks, welche lange nicht mehr beobachtet wurden, gelöscht. Mit der Feststellung, dass sich die Kovarianz von Hypothesen, denen keine Beobachtung assoziiert wird, mit jeder Anwendung des Bewegungsmodells im Bayes-Filters erhöht (Abschnitt 4.5), kann das Löschen auf Basis von einer hohen Kovarianz in der Position erfolgen:

$$\mathbf{\Sigma}_{ij} > T_{Position} \; . \tag{4.16}$$

Hierbei geben ij die Elemente des Positionsanteils der Kovarianzmatrix $\mathbf{\Sigma}_{(x,y,z)}$ an und $T_{Position}$ stellt einen Schwellwert zur Löschung dar.
Zusätzlich werden Hypothesen mit geringer Existenzwahrscheinlichkeit gelöscht:

$$p(H_{Person}) < T_{Existenz} \; . \tag{4.17}$$

Dabei handelt es sich vor allem um Hypothesen, die zuvor in Hindernissen lagen (siehe Abschnitt 4.9.2) und deren Existenzwahrscheinlichkeit dadurch verringert wurde. Als letzten Mechanismus fusioniert der Personentracker Hypothesen, welche sehr nah beieinanderliegen und einen ähnlichen Geschwindigkeitsvektor aufweisen. Hierdurch werden duplizierte Tracks beseitigt.

4.9.2. Nutzen von Umgebungswissen

Der Personentracker erlaubt es Wissen aus der Umgebung in Form von Umgebungskarten (Abschnitt 2.3.1 und Abbildung 4.1) zu nutzen, um die Verarbeitung von Beobachtungen einzuschränken und die Plausibilität von Hypothesen zu überprüfen.

Ignorierte Bereiche in Umgebungskarten

In der Umgebungskarte können Bereiche händisch oder dynamisch markiert werden, an denen der Roboter keine Beobachtungen durchführen soll. Dabei werden alle Beobachtungen, die in markierten Zellen der Umgebungskarte liegen, ignoriert. Dies kann einerseits genutzt werden, um Bereiche, in denen der Roboter dem Nutzer nicht folgen beziehungsweise beobachten soll, auszuschließen (z. B. Badezimmer). Andererseits kann dieser Mechanismus genutzt werden, um Detektionen und Hypothesen bei zusätzlichem Wissen als falschpositiv zu markieren (Abschnitt 5.3).

Umgebungskarten zur Hypothesenvalidierung

Die Belegtheitsinformation der Umgebungskarte wird verwendet, um Hypothesen in Hindernissen und Wänden abzuschwächen. Dabei wird die Existenzwahrscheinlichkeit dieser Hypothesen mit einem konstanten Faktor pro Zeitschritt verringert, bis sie den Wert 0 erreicht. Durch den zuvor beschriebenen Löschmechanismus werden diese Hypothesen anschließend aus der Trackliste entfernt (Abschnitt 4.9.1).

4.10. Experimentelle Untersuchungen

Dieser Abschnitt beschreibt die experimentelle Evaluation des Personentrackers mit den zugehörigen Detektionsmodulen. Zunächst werden kurz die Ergebnisse aus Volkhardt u. a. (2013a) zusammengefasst. Anschließend wird die Evaluation der Detektoren und des Personentrackers vorgestellt. Zum Abschluss werden Experimente im realen Anwendungsszenario präsentiert.

4.10.1. Evaluation aus Volkhardt u. a. (2013a)

In Volkhardt u. a. (2013a) wurde der Personentracker auf 8 öffentlichen Datensätzen evaluiert (Anhang A.7.1). Die Datensätze umfassen unter anderem einen stehenden, fahrenden, folgenden und suchenden Roboter mit mehreren Personen. Während der Evaluation wurde ein echtzeitfähiger Personentracker unter Nutzung eines Gesichts-, HOG-, Oberkörper-HOG-, Bewegungs- und Beinpaardetektors mit einem rein Beinpaardetektionsbasierten Tracker verglichen. Zum Zeitpunkt der Veröffentlichung lagen der DPM- und der FPDW-Detektor (Abschnitte 3.3.4 und 3.3.5) noch nicht in einer echtzeitfähigen Implementierung vor und konnten daher nur in offline Varianten untersucht werden. Eine ausführlichere Darstellung der Ergebnisse findet sich in Anhang A.10.1 beziehungsweise Volkhardt u. a. (2013a).
Zusammenfassend zeigte sich, dass der Personentracker gute Resultate zeigt, wenn sich Personen in aufrechter Pose befinden; also stehen oder laufen. Bei sitzenden Personen nimmt die Performance jedoch stark ab. Die Fusion mehrerer Detektionsmodule ist – wie zu erwarten – einem rein beinpaarbasierten Tracker überlegen. Der offline FPDW-basierte Tracker zeigte nur gute Resultate, wenn sich Personen in aufrechter Pose befinden, da der Detektor speziell für Fußgänger im Straßenverkehr trainiert wurde. Die besten Resultate wurden auf nahezu allen Datensätzen mit einem offline DPM-basierten Personentracker erzielt. Generell konnte die Performance verbessert werden, indem zusätzlich Beinpaardetektionen in den jeweiligen Personentracker fusioniert wurden.

Die Datensätze mit sitzenden Personen zeigten, dass die verwendeten Detektoren nicht ausreichend sind, um sitzende Personen robust zu tracken. Im Verlauf dieser Dissertation konnten echtzeitfähige Implementierungen des FPDW- und DPM-Detektors genutzt werden. Die dadurch entstandenen Verbesserungen gegenüber den Ergebnissen aus diesem Abschnitt werden unter anderem in den folgenden Abschnitten behandelt.

4.10.2. Konzeption der Evaluation

Im Gegensatz zu Volkhardt u. a. (2013a) wird der Personentracker im Folgenden mit allen bisher beschriebenen Komponenten und Funktionalitäten und unter Verwendung des FPDW- und des DPM Detektors evaluiert. Auf den Einsatz des Gesichts- und Bewegungsdetektors wurde verzichtet, da diese keinen Mehrwert lieferten (Abschnitt 4.10.3).

Datensätze

Die Evaluation der Detektoren und des Personentrackers erfolgt auf insgesamt 16 Datensätzen. Dabei werden die 8 Datensätze aus Volkhardt u. a. (2013a) (Anhang A.7.1) und 8 zusätzlichen Datensätzen mit erhöhter Schwierigkeit (Anhang A.7.2) in häuslicher Umgebung verwendet. In den Letztgenannten wurde besonders auf realistische Bedingungen geachtet, indem die Wohnung mit zusätzlichen Gegenständen und Möbelstücken ausgestattet wurde und natürliche Beleuchtungsbedingungen, wie beispielsweise Tageslicht, Stehlampen und zugezogenen Rollos, vorherrschten. Zur Übersichtlichkeit wurden die Datensätze nach ihrem Inhalt in vier Gruppen eingeteilt.

- **Stehend:** In diesen Datensätzen steht oder fährt der Roboter während sich bis zu 4 Personen in seinem Sensorsichtbereich aufrecht bewegen.

- **Folgend:** Diese Datensätze umfassen ein Folgenszenario, bei dem der Roboter dem Nutzer durch die Wohnung folgt, während andere Personen den Weg kreuzen.

- **Sitzend:** Hier betritt eine Person den Sichtbereich des Roboters, setzt sich und verlässt den Sichtbereich anschließend wieder.

- **Suche:** Den Großteil der Datensätze umfassen Suchszenarien, bei denen der Roboter einen oder mehrere unterschiedliche Nutzer in der Wohnung sucht. Diese beinhalten hauptsächlich sitzende, aber auch einige stehende, Posen.

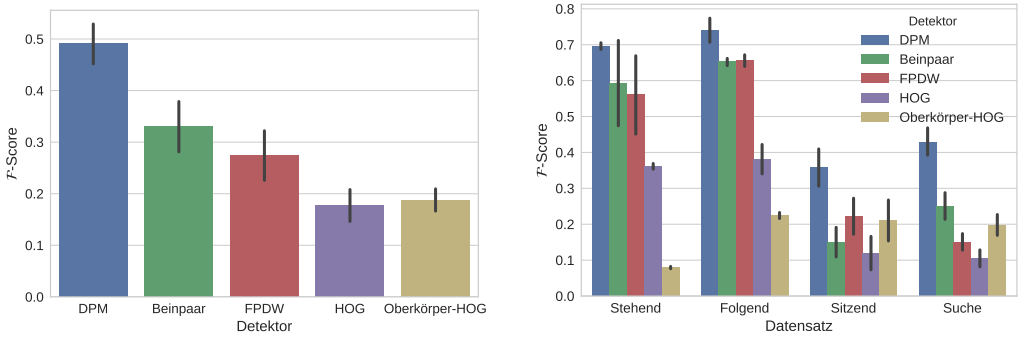
Weitere Informationen, wie Inhalt, Länge und Anzahl der Bilder, sowie ein visueller Eindruck der Datensätze finden sich in Anhang A.7.2. Alle Personen in den Datensätzen sind händisch mit Bounding-Boxen, IDs und Verdeckungsinformation mithilfe des VATIC Label-Tools annotiert (Carl Vondrick 2012).

Auswertungsmethodik und Bewertungsmetriken

Zur Auswertung werden alle gelabelten Bounding-Boxen zu Posen in Weltkoordinaten transformiert (Anhang A.4.1). Unter Berücksichtigung von Verdeckungen wird die Ground-Truth mit den Detektor- bzw. Trackerhypothesen und den zugehörigen IDs verglichen. Die Bewertung ist dabei analog zu Anhang A.10.1 mit dem Unterschied, dass die Evaluation auf Basis von Posen statt Bounding-Boxen erfolgt. Als Abstandsmaß für Detektion und Ground-Truth wird die euklidische Distanz mit einem Schwellwert von 0.5 m verwendet. Als Evaluationsmetriken kommen der $\mathcal{F}$-*Score* (harmonisches Mittel aus *Recall* und *Precision*) und die *Multiple Object Tracking Performance* (MOTP) zur Anwendung (Anhang A.9). Dadurch lassen sich die Präzision, Genauigkeit und Konsistenz des Trackers auf den unterschiedlichen Szenarien auswerten und vergleichen. Zur besseren statistischen Abschätzung wurden alle Experimente auf jedem Datensatz fünfmal wiederholt und die Ergebnisse statistisch gemittelt.

4.10.3. Ergebnisse der Personendetektion

Dieser Abschnitt beschreibt die Evaluation der eingesetzten Detektoren gegenüber den gelabelten Bounding-Boxen. Abbildung 4.6(a) zeigt den mittleren $\mathcal{F}$-Score der Detektoren aller Versuche auf allen verwendeten Datensätzen. Die schlechtesten Ergebnisse werden mit den HOG Detektoren erreicht. Der FPDW und der Beinpaardetektor liefern signifikant bessere $\mathcal{F}$-Scores. Die mit Abstand besten Resultate werden mit dem DPM Detektor erreicht. Hauptgrund ist die komplexe Modellierung der Person mit Körperteilen (Abschnitt 3.3.4).
Zum tieferen Verständnis der Stärken und Schwächen der Detektoren listet Abbildung 4.6(b) eine Auswertung nach dem Inhalt der Datensätze. Allgemein weisen die *Sitzend* und *Suche* Datensätze für alle Detektoren höhere Falsch-positive auf, die meist durch Schränke, Tische, Stühle oder Stehlampen verursacht werden. Der Oberkörper-HOG Detektor liefert für aufrechte, meist weit entfernte Personen schlechte Ergebnisse, kann aber bei sitzenden Posen ähnliche und teilweise bessere Ergebnisse als der HOG- und Beinpaardetektor liefern.
Der HOG- und der FPDW-Detektor verhalten sich ähnlich, wobei der FPDW-Detektor stets signifikant bessere Ergebnisse erzielt. Bei stehenden Personen



(a) $\mathcal{F}$-Score **(b)** $\mathcal{F}$-Score nach Datensatz

Abbildung 4.6.: Mittlerer $\mathcal{F}$-Score der verwendeten Detektoren auf (a) allen Datensätzen und (b) nach Datensätzen gruppiert. Die Fehlerbalken geben das 95% Konfidenzintervall an.

werden gute Ergebnisse erzielt, die den Ergebnissen des DPM Detektors nahekommen. Bei sitzenden Posen sinkt die Performance erheblich.
Der Beinpaardetektor erreicht einen guten $\mathcal{F}$-Score, wenn die Beine der Person sichtbar sind. Dies ist vor allem bei aufrechten Posen aber auch in einigen Suchszenarien der Fall. Vor allem die Verbesserungen aus Abschnitt 3.2.2 bewirken, dass der Beinpaardetektor die zweitbesten Ergebnisse liefern kann.
Der DPM Detektor erzielt auf allen Datensätzen die besten Ergebnisse. Zwar sinkt die Performance wie bei den anderen Detektoren, wenn der Nutzer sitzt, der mittlere $\mathcal{F}$-Score liegt aber weit über allen anderen Detektoren.
Die Auswertung legt nahe, dass der Personentracker am sinnvollsten mit dem DPM Detektor betrieben werden sollte. Die vorgestellten Erweiterungen aus Abschnitt 3.3.4 ermöglichen den echtzeitfähigen Einsatz des Detektors. Eine Fusion mit dem Beinpaardetektor ist in jedem Fall vorteilhaft (Volkhardt u. a. 2013a).

Gesichts- und Bewegungsdetektor

Der Gesichts- und Bewegungsdetektor aus Volkhardt u. a. (2013a) lieferten im Vergleich zu den anderen Detektoren unzureichende Ergebnisse, weil sie den Nutzer einerseits nur in bestimmten Situationen erkennen – wenn das Gesicht sichtbar ist, beziehungsweise wenn der Nutzer in Bewegung ist und der Roboter stillsteht – und andererseits hohe Falsch-positiv-Raten aufweisen. Der Gesichtsdetektor verschlechterte das Tracking eher, als es zu verbessern. Verfahren, wie Kublbeck u. a. (2006) und Zhang u. a. (2010), könnten hier einen Zugewinn darstellen. Als aussichtsreicher Kandidat bietet sich das Verfahren

Tabelle 4.1.: Rechenzeit der Module

Modul	Mittlere Rechenzeit [ms]
Gesichtsdetektor	99.7
Oberkörper- / HOG-Detektor	242.3 / 225.4
Bewegungs- / Beinpaardetektor	1.6 / 1.0
FPDW	142.1
DPM (4 Kerne, sep. CPU)	265.3
Personentracker	0.2

von Mathias u. a. (2014) an, welches Gesichtsmodelle auf Basis von P. F. Felzenszwalb u. a. (2010) und Dollar u. a. (2010) benutzt. Der Bewegungsdetektor lieferte ebenfalls hohe Falsch-positiv-Raten, vor allem bei Fernsehern und Fenstern. Aufgrund der Übersichtlichkeit wurde in den obigen Ausführungen auf die Evaluation des Gesichts- und Bewegungsdetektors verzichtet. Die Detektoren werden aus den oben genannten Gründen ebenfalls nicht im Personentracker und Anwendungsszenario verwendet.

Performance der Detektoren

Die Rechenzeit der Detektoren für die Verarbeitung eines Bildes bzw. Laserscans ist in Tabelle 4.1 angegeben. Da die Rechenzeit der einzelnen Detektoren relativ hoch ist, können nicht alle Module gleichzeitig mit dem Personentracker verwendet werden.
An dieser Stelle wird auch die relativ hohe Rechenzeit des Gesichtsdetektors ersichtlich. Für den DPM Detektor wurde eine separate CPU mit 4 Kernen eingesetzt, während die anderen Detektoren gemeinsam auf dem Hauptprozessor des Roboters arbeiten. Aus der Tabelle wird ebenfalls ersichtlich, dass die visuellen Detektoren nicht jedes Frame der Kamera verarbeiten können. Ein Personentracker bringt also Vorteile, um die Hypothese zwischen den einzelnen verarbeiteten Bildern per Prädiktion und Beinpaardetektionen weiter zu tracken.

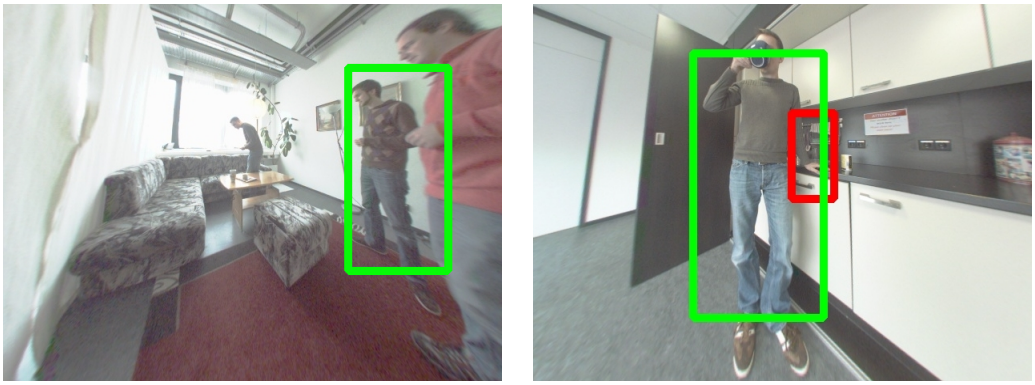
4.10.4. Ergebnisse des Personentrackings

Der Personentracker wurde mit unterschiedlichen Detektorkombinationen auf den Datensätzen evaluiert. Dabei wurde das lineare Systemmodell mit konstanter Geschwindigkeit (Volkhardt u. a. 2013a) mit einem EKF[9] verwendet.

[9]Durch die Verwendung eines linearen Systemmodells entspricht der EKF einem linearen Kalman-Filter.

(a) Aufrechte Posen **(b)** Sitzender Nutzer

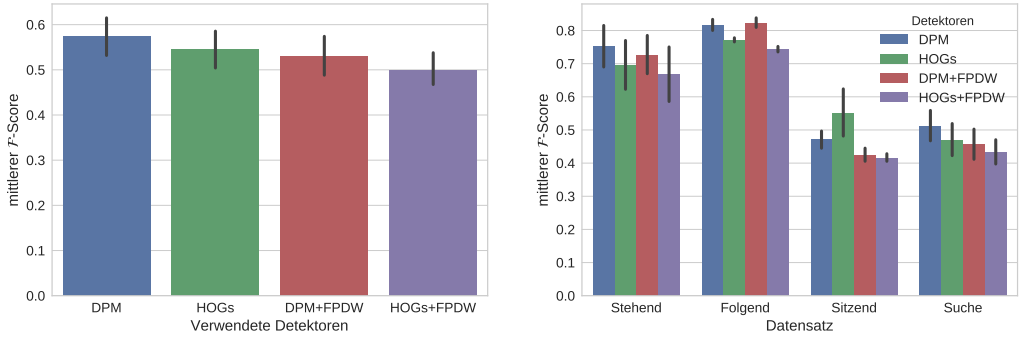
Abbildung 4.7.: Qualitative Trackingergebnisse. (a) Tracking mehrerer Personen unter Verdeckungen. (b) Tracking bei schlechten Beleuchtungsverhältnissen.



(a) Falsch-negative **(b)** Falsch-positive

Abbildung 4.8.: Qualitative Trackingprobleme. (a) Zwei Personen werden aufgrund von teilweisen Verdeckungen nicht erkannt. (b) Es treten falsch-positiv Hypothesen in stark strukturierten Hintergründen auf (rote Box).

Der Beinpaardetektor wurde aufgrund der guten Ergebnisse und der geringen Rechenzeit immer eingesetzt. Als visuelle Detektoren kommen jeweils der DPM Detektor und die beiden HOG Detektoren zum Einsatz. Beide werden in weiteren Kombinationen mit dem FPDW Detektor verwendet.
Abbildung 4.7 zeigt exemplarisch qualitative Ergebnisse des Personentrackings in schwierigen Situationen. In Abbildung 4.7(a) werden mehrere Personen in unterschiedlichen Distanzen und Posen unter Verdeckungen getrackt. Abbildung 4.7(b) zeigt eine Szene, in der der Nutzer auch bei schwachem Licht

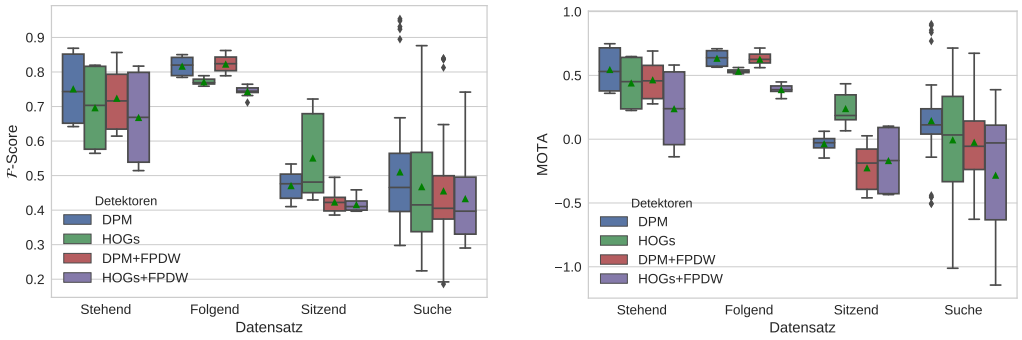


(a) $\mathcal{F}$-Score **(b)** $\mathcal{F}$-Score nach Datensatz

Abbildung 4.9.: Quantitative Resultate. (a) erreichter $\mathcal{F}$-Score für verschiedene Detektorkombinationen. (b) $\mathcal{F}$-Score aufgeschlüsselt nach Inhalt der Datensätze. Die senkrechten Linien geben das 95% Konfidenzintervall für die Mittelwertschätzung an.

erkannt wird. Aufgrund der verdeckten Beinpaare ist die Tiefenschätzung der Hypothese und damit auch die Größe der Bounding-Box leicht fehlerhaft. Abbildung 4.8 zeigt einige Problemfälle des Trackings. In Abbildung 4.8(a) werden zwei Personen nicht erkannt, da diese teilweise verdeckt sind. In Abbildung 4.8(b) wird der Nutzer getrackt, es tritt jedoch auch eine falsch-positiv Hypothese auf.
Abbildung 4.9(a) zeigt den erreichten mittleren $\mathcal{F}$-Score auf allen Datensätzen. Für alle Detektorkombinationen sind die erreichten Werte höher als die der Einzeldetektoren aus Abschnitt 4.10.3. Dies verdeutlicht, dass die Vorteile eines Personentrackers (Abschnitt 4.1) zu einem besseren Ergebnis führen. Die besten Ergebnisse werden mit dem DPM Detektor erreicht. Ein Personentracker mit diesem visuellen Detektor übertrifft einen Tracker mit HOG und Oberkörper-HOG Detektor. Die Hinzunahme des FPDW Detektors führt in beiden Fällen zu einem Absinken des $\mathcal{F}$-Scores. Gründe hierfür sind Falsch-positive des Detektors und fehlende zusätzliche Richtig-positive[10].
Abbildung 4.9(b) gruppiert den $\mathcal{F}$-Score nach dem Inhalt der Datensätze. Hier ergibt sich ein ähnliches Bild wie bei den Einzeldetektoren in Abbildung 4.6(b). In Datensätzen mit aufrechten Posen ist die Performance des Personentrackers höher, als in Datensätzen mit hauptsächlich sitzenden Nutzern. Allerdings erkennt man auch hier klar die Vorteile der Detektorfusion, sodass der $\mathcal{F}$-Score

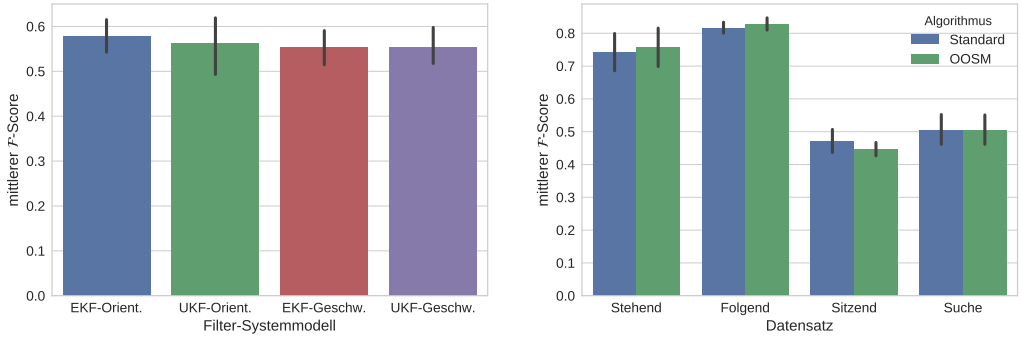
[10]Der DPM und die HOG Detektoren erkennen stehende Personen meist in den gleichen Fällen wie der FPDW Detektor. Zusätzlich werden diese Situationen aber oftmals schon durch den Beinpaardetektor erfasst.



(a) $\mathcal{F}$-Score **(b)** MOTA

Abbildung 4.10.: Boxplots der Metriken gruppiert nach dem Inhalt der Datensätze. Die Boxplots beschreiben die Verteilung der Metriken mit Median, oberes und unteres Quartil, unterer und oberer Whisker sowie dem Mittelwert (grüne Dreiecke).

der Tracker über dem der Einzeldetektoren liegt. Erwähnenswert sind die guten Resultate des HOG-Trackers auf den *Sitzend* Datensätzen. Hier verursachen die HOG Detektoren sehr viel weniger Falsch-positive als der DPM Detektor bei nahezu gleicher Richtig-positiv-Rate, was zu einem besseren $\mathcal{F}$-Score führt. Zum tieferen Verständnis sind in Abbildung 4.10 die Verteilungen des $\mathcal{F}$-Scores und der MOTA auf den Datensätzen als Boxplots dargestellt. Im Vergleich zu Abbildung 4.9(a) wird in Abbildung 4.10(a) deutlich, dass die Performance des Personentracks je nach Datensatz merklich schwankt. So werden bei den *Suche* Datensätzen und dem DPM-Personentracker beispielsweise $\mathcal{F}$-Scores im Bereich von $[0.30, 0.95]$ erzielt. Ursache hierfür sind die stark schwankenden Posen der einzelnen Testnutzer und die Beleuchtungsbedingungen. Selbst bei aufrechter Pose im Fall der *Stehend* Datensätze sind die Unterschiede erkennbar, da einige Datensätze aus Volkhardt u. a. (2013a) (Anhang A.7.1) einfachere Bedingungen für den Tracker enthalten als die „realistischeren“ Datensätze dieser Arbeit (Anhang A.7.2).
Die MOTA in Abbildung 4.10(b) zeigt im Sinne der Rangliste ein ähnliches Bild wie der $\mathcal{F}$-Score. Im relativen Vergleich sind die Werte geringer, da die MOTA neben Falsch-positiven und Falsch-negativen auch noch die Konsistenz der Tracks bewertet. Dabei werden die IDs der Tracks mit denen der GroundTruth verglichen und Abweichungen bestraft. Abweichungen können durch Abreißen und Neugenerierung von Tracks, aber auch durch Vertauschungen von nahegelegenen Tracks, entstehen. Die Nutzung des DPM-Detektors fördert demnach auch die Konsistenz der Hypothesen-Tracks im Personentracker. Die exakten numerischen Werte der quantitativen Evaluation dieses Abschnittes finden sich



(a) Filter und Systemmodelle **(b)** OOSM

Abbildung 4.11.: Filter, Systemmodelle und OOSM. (a) zeigt die erzielten $\mathcal{F}$-Scores mit unterschiedlichen Filtern und Systemmodellen. Die erreichten Ergebnisse sind ähnlich, wobei das Modell mit konstanter Orientierung und Geschwindigkeit in Kombination mit dem EKF die besten Ergebnisse erzielt. (b) vergleicht den Personentracker mit und ohne Behandlung von veralteten Beobachtungen (OOSM). Die Fehlerbalken geben das 95% Konfidenzintervall der Mittelwertschätzung an.

in Anhang A.10.2.

Filter, Systemmodelle und OOSM

In Arenknecht (2015) wurden die verschiedenen Filter und Systemmodelle evaluiert. Es zeigte sich, dass kein sichtbarer Unterschied zwischen dem EKF und dem UKF vorhanden ist. Weiterhin erzielt das Modell mit konstanter Orientierung und Geschwindigkeit (Bellotto u. a. 2009) minimal bessere Ergebnisse als das Modell mit konstanter Geschwindigkeit (Volkhardt u. a. 2013a). Die Ergebnisse sind in Abbildung 4.11(a) zusammengefasst. Für einen weiterführenden Vergleich sei auf Arenknecht (2015) verwiesen. Aufgrund der einfacheren Berechenbarkeit wurde sich für die Nutzertests und den Einsatz im Anwendungsszenario für das lineare Systemmodell mit konstanter Geschwindigkeit mit einem EK-Filter entschieden. Als weiterer Vorteil ermöglicht das lineare Modell die Rückwärtsprädiktion zur Behandlung von Beobachtungen außer der Reihe (Abschnitt 4.7.2). Diese findet in allen Fällen statt. In den Experimenten zeigte sich, dass die Rückwärtsprädiktion vor allem in Datensätzen mit sich bewegenden Personen leicht bessere Resultate gegenüber keiner Behandlung von veralteten Beobachtungen erzielt. Bei sitzenden oder stehenden Personen waren die Ergebnisse in beiden Fällen ähnlich. Die entsprechenden $\mathcal{F} - Scores$ sind in Abbildung 4.11(b) dargestellt.

Tabelle 4.2.: Übersicht über Testläufe zum Folgeverhalten (Vgl. Gross u. a. (2015)).

Umgebung	Anzahl	Erfolgsrate
WhgM1	48	0.87
WhgM2	65	0.94
WhgM3	72	0.85
WhgS1	18	1.00
WhgS2	9	0.67
WhgS3	27	0.85
WhgS4	38	0.92
WhgS5	27	0.89
WhgS6	26	0.96
WhgS7	25	0.96
WhgS8	27	0.89
WhgS9	12	1.00

Performance des Personentrackers

Das gesamte Trackingsystem läuft in Echtzeit auf den on-board Prozessoren des Roboters (2 Intel i7-620M quad core). Den Hauptanteil der Rechenzeit benötigen die Detektoren (Abschnitt 4.10.3). Dabei wird 1 Prozessor fast ausschließlich für den DPM-Detektor verwendet. Für die anderen Detektoren und den Personentracker stehen etwa 60% der Prozessorleistung zur Verfügung, um genügend Ressourcen für die anderen benötigten Funktionalitäten des Roboters wie Lokalisierung, Navigation, Ablaufsteuerung und Nutzerdialog (Gross u. a. 2015) bereitzustellen. Der Personentracker selbst, benötigt nur eine mittlere Rechenzeit von 0.2 ms und ist damit das performanteste Modul.

4.10.5. Evaluation im realen Szenario: Folgeverhalten

Der Personentracker mit den zugehörigen Detektoren wurde nicht explizit quantitativ in den Seniorenwohnungen untersucht. Das System wurde jedoch implizit in verschiedenen Anwendungen, wie der Personensuche (siehe Abschnitt 5.4.3) oder dem Folgen des Nutzers evaluiert. An dieser Stelle werden die Ergebnisse des Folgeverhaltens aus Gross u. a. (2015) präsentiert, da für das robuste und schnelle Folgen des Nutzers eine kontinuierliche Personenhypothese benötigt wird.
Tabelle 4.2 zeigt einen Ausschnitt der Ergebnisse aus Experimenten in vier Mitarbeiter- und acht Seniorenwohnungen (Anhang A.8). In jedem Versuch wurden mehrere Start- und Endpositionen festgelegt, zwischen denen der Roboter geführt wurde. Dabei wurde gemessen, wie oft der Roboter dem Nutzer zur Endposition folgte. Ein Misserfolg wurde gezählt, wenn der Roboter den

Nutzer verloren hatte oder mit einem Hindernis kollidierte. Die hohe Erfolgsrate in fast allen Versuchen zeigt, dass der Personentracker aufrechte Personen auch im realen Anwendungsszenario kontinuierlich tracken kann. Signifikante Unterschiede bei unterschiedlichen Beleuchtungsbedingungen oder der Komplexität der Wohnung konnten nicht festgestellt werden. Die relative schlechte Erfolgsrate in Wohnung *WhgS2* war auf eine kontinuierliche falsch-positiv Hypothese, verursacht durch einen Tisch, zurückzuführen. Weitere Ergebnisse und Gütemaße finden sich in Gross u. a. (2015). Zum Vergleich nutzen andere Evaluationen von Folgeverhalten weit weniger quantitative Maße (Xudong u. a. 2008; PR2 2010; Cosgun u. a. 2013).

4.11. Diskussion und Fazit

Dieses Kapitel stellte den entwickelten echtzeitfähigen, multi-modalen Personentracker für den mobilen Roboter vor. Neben einer Erläuterung verschiedener Filterverfahren und der eingesetzten Systemmodelle wurde auf die Besonderheiten der softwareseitigen Umsetzung eingegangen. Diese ermöglicht die einfache Kombination von verschiedenen Filtern und Systemmodellen. Der Personentracker wurde beispielsweise mit angepassten Systemmodellen und zusätzlichen Detektoren (Weinrich u. a. 2012) erfolgreich zum Tracken der *Blickrichtung* von Personen eingesetzt (Weinrich u. a. 2013b).
Die Datenassoziation erfolgt im Zustandsraum und verwendet Covariance Intersection für abhängige Beobachtungen eines Sensors. Beobachtungen, die am Personentracker verspätet und außer der Reihe ankommen, werden mittels Rückwärtsprädiktion (Retrodiction) integriert. Neben der eigentlichen Filterung schätzt der Personentracker für jede Hypothese eine Existenzwahrscheinlichkeit mithilfe eines Naïve-Bayes Klassifikators. Ferner wurden das Hypothesenmanagement und die Nutzung von optionalem Wissen in Form von Umgebungskarten beschrieben.
Die Evaluation untersuchte zunächst die Eignung der Detektoren auf realistischen Datensätzen. Anschließend wurde der Personentracker mit verschiedenen Kombinationen der Detektoren evaluiert und der Beinpaardetektor in Kombination mit dem DPM-Detektor für die Nutzertests ausgewählt. Daraufhin erfolgte eine Untersuchung unterschiedlicher Filter und Systemmodelle sowie der Behandlung von asynchronen Beobachtungen außer der Reihe.
Zum Abschluss wurden als Anwendungsfall des Personentrackers Ergebnisse des Folgeverhaltens aus den Experimenten in den Seniorenwohnungen präsentiert. In den Experimenten zeigte sich, dass der Personentracker robust genug arbeitet, um die Module der folgenden Kapitel zu ermöglichen (Kapitel 5) und es dem Roboter erlaubte, mehrere Tage im realen Einsatzszenario autonom

mit Senioren zu interagieren (Kapitel 7). Allerdings zeigten sich auch Unzulänglichkeiten des Trackingsystems, was sich vor allem in Falsch-negativen bei schwierigen Posen und Falsch-positiven bei personenähnlichen Objekten niederschlägt. Auf Basis von konsistenten Personentracks lassen sich, neben den in dieser Arbeit vorgestellten, auch weitere Funktionalitäten entwickeln. Beispiele umfassen die Aktivitätsschätzung (Volkhardt u. a. 2010) und die Gang- und körperliche Fitnessanalyse (Stiebritz 2014)[11].

[11] Vom Autor im Rahmen dieser Arbeit betreut.

5. Personensuche

Dieses Kapitel beschreibt das entwickelte Modul zur Personensuche. Die Arbeit stellt drei Verfahren vor, die historisch entstanden sind und eine zunehmende Leistungsfähigkeit an Genauigkeit und Geschwindigkeit besitzen.

5.1. Einleitung

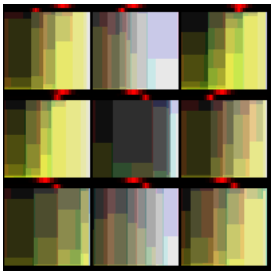
Eine Nutzersuche muss immer dann vom Roboter initiiert werden, wenn der Nutzer den limitierten Erfassungsbereich der Sensorik verlassen hat und seine Interaktion benötigt wird, z. B. bei Terminerinnerungen oder eingehenden Anrufen. Zusätzlich kann der Nutzer den Roboter bei Bedarf mittels eines Knopfes auf einer Fernbedienung zu sich rufen. Während der Suchfahrt muss der Roboter den Nutzer in verschiedenen Posen, z. B. stehend und sitzend erkennen. Dabei soll der Roboter den Nutzer rasch in der Wohnung finden, robust wahrnehmen und sich ihm auf Interaktionsdistanz nähern. Da das Trackingsystem nicht perfekt ist, muss das System mit falsch-positiv Detektionen umgehen.

5.2. Suche von Personen an typischen Aufenthaltsorten

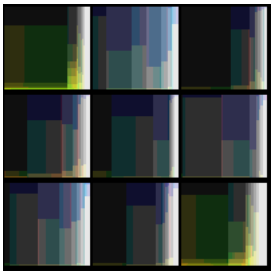
In Volkhardt u. a. (2011a,b) erkennt der Roboter stehende Personen während der Suchfahrt mit einer vorläufigen Trackingarchitektur, ähnlich zu Abschnitt 2.3. Zur Zeit der Veröffentlichung der Verfahren stellte die Erkennung sitzender Personen auf einem mobilen Roboter eine große Herausforderung dar, da fortgeschrittene Detektionsverfahren, wie das DPM (Abschnitt 3.3.4) noch nicht auf dem Roboter verfügbar waren. Daher modellieren Volkhardt u. a. (2011a,b) zunächst typische Aufenthaltsorte, wie Couch, Sessel oder Stühle, in der Wohnung.

5.2.1. Detektion von Personen an Aufenthaltsorten

Im Vorfeld der Erkennung wird ein visuelles Modell der unbesetzten Aufenthaltsorte (Plätze) aufgebaut. Die visuellen Modelle bestehen aus kontextuellen



(a) Platzmodell



(b) Nutzermodell

(c) Beleuchtungen

Abbildung 5.1.: (a) Platzmodell mit neun Farbhistogrammen und zugehöriger 2D-Kontextverteilung (rote Linien). (b) Nutzermodell mit neun Farbhistogrammen. (c) Platz unter unterschiedlichen Beleuchtungen sowie Nutzer mit unterschiedlicher Kleidung (Volkhardt u. a. 2011b).

Farbhistogrammen, welche die Erscheinung der Plätze in jeweils acht Bins des RGB-Farbraums mit zusätzlichen Kontextinformationen über die Aufnahme erfassen. Zur Generierung werden die Plätze in der Wohnung vorab als 3D-Boxen definiert, in das Kamerabild projiziert und vom Roboter aus mehreren festgelegten Navigationspunkten betrachtet. Dabei wird ein multi-modales Farbhistogramm H_i aufgebaut. Die Bedingungen der Aufnahme, wie Blickrichtung und Tageszeit, werden in einer zugehörigen diskreten Kontextverteilung C_i gespeichert (Abbildung 5.1(a)). Das Modell $\mathcal{M} = \{\kappa_1, \ldots, \kappa_n\}$, mit $\kappa_i = (H_i, C_i)$ nutzt n distinkte Komponenten κ_i, während ähnliche Ansichten und Kontexte fusioniert werden. Die Ähnlichkeit s zweier Komponenten bestimmt sich mit:

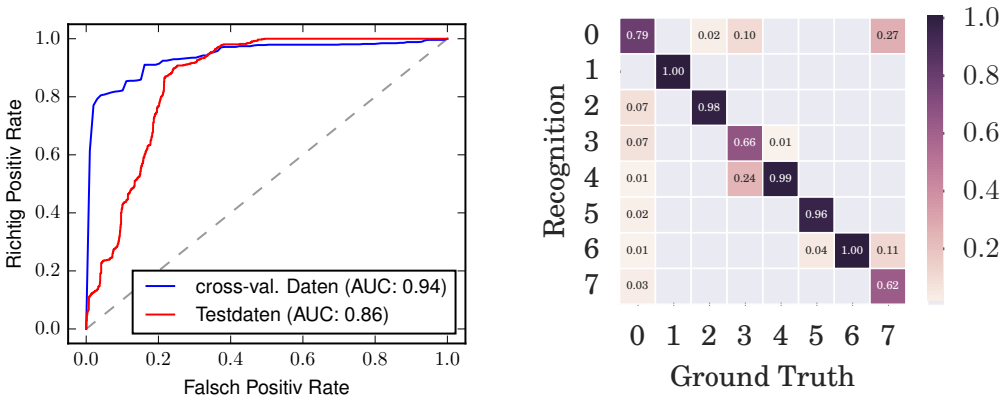
$$s = \mathcal{BC}\left([H_i, C_i], [H_j, C_j]\right) \ , \tag{5.1}$$

wobei $[H, C]$ die Konkatenation der Histogramm- und der Kontextverteilung bezeichnet und $\mathcal{BC}(p, q)$ den von Bhattacharyya (1943) vorgestellten Bhattacharyya Koeffizienten[1] darstellt:

$$\mathcal{BC}(p, q) = \sum_{x \in X} \sqrt{p(x)q(x)} \ . \tag{5.2}$$

Zusätzlich wird ein visuelles Modell des Nutzers mittels Hintergrundsegmentierung auf Basis von Gaussian Mixture Modellen (Stauffer u. a. 1999) und dem GrabCut Algorithmus (Rother u. a. 2004) erfasst (Abbildung 5.1(b)). Dieses entspricht dem Farbhistogramm der Platzmodelle ohne die Kontextverteilung.

[1] Der Bhattacharyya Koeffizient wird in der Statistik eingesetzt, um die Ähnlichkeit von diskreten oder kontinuierlichen Verteilungen beziehungsweise zweier Samples zu bestimmen.



(a) ROC Kurven **(b)** Konfusionsmatrix

Abbildung 5.2.: (a) ROC Kurven für cross-validierte und unabhängige Testdaten. (b) Konfusionsmatrix mehrerer Testdurchläufe. Klassen 1-7 repräsentieren unterschiedliche Aufenthaltsorte. Bei Klasse 0 befindet sich der Nutzer nicht im Apartment (Volkhardt u. a. 2011a).

Während der Suche fährt der Roboter die einzelnen Navigationspunkte ab und entscheidet mittels Gleichung (5.1) und einem SVM-Klassifikator, ob die aktuelle Ansicht des Aufenthaltsortes oder Teile davon eher zum leeren Platzmodell oder zum Personenmodell passen und der Platz somit belegt ist.

5.2.2. Ergebnisse und Bewertung

Das Verfahren wurde für sieben Aufenthaltsorte in einem bewohnten 60 m^2 3-Raum-Apartment zu unterschiedlichen Tageszeiten evaluiert (siehe Abbildung 5.1(c)). Der Roboter untersucht jeden Platz und speichert die Ähnlichkeit der Ansicht zum zuvor gelernten Nutzer- bzw. Platzmodell. Eine lineare SVM wurde auf den gesammelten Ähnlichkeiten der Plätze trainiert. Die tatsächliche Nutzerposition wurde manuell gelabelt. Die Klassifikationsergebnisse dieser SVM finden sich in Abbildung 5.2(a).
In Abbildung 5.2(b) ist eine Konfusionsmatrix dargestellt, bei der jeder Platz als Klasse 1-7 repräsentiert wird; mit Label 0 als Kennzeichnung, dass sich der Nutzer nicht in der Wohnung befindet. Die durchschnittliche Klassifikationsrate liegt bei über 85 %, wobei einige Verwechslungen und Falsch-Detektionen auftreten. Die Hauptgründe liegen in überlappenden Ansichten der Orte im Kamerabild und starken Beleuchtungsschwankungen, die eine korrekte Farbextraktion erschweren. Weitere Details und Ergebnisse finden sich in Volkhardt u. a. (2011a).

Tabelle 5.1.: Ergebnisse der Suchdurchläufe (Volkhardt u. a. 2011b).

	Suchfahrten	Erfolgsrate	durchschn. Zeit
Sitzend/liegend/stehend	73	0.74	27.8 s
Stehend	15	0.87	32.2 s
Ohne Smart-Home Sens.	19	0.44	37.2 s
Nutzer nicht in Wohnung	18	0.15	25.4 s

In Volkhardt u. a. (2011b) wurde das Verfahren mittels externer Smart-Home Bewegungssensoren verbessert und in über 100 Suchfahrten evaluiert. Tabelle 5.1 zeigt die Verbesserung der Erfolgsrate, wenn der Roboter durch die Smart-Home Sensoren bereits den Raum beziehungsweise Raumteil kennt, in dem sich die Person zuletzt aufgehalten hat. Die geringe Erfolgsrate, falls sich der Nutzer nicht in der Wohnung befindet, ist dem geschuldet, dass der Roboter alle Aufenthaltsorte der Wohnung untersucht und an jedem falsch-positiv Detektionen auftreten können, welche nicht gesondert behandelt werden.
Beide Verfahren ermöglichten dem Roboter zum Stand 2011 erstmals die Erkennung sitzender und liegender Personen während der Suchfahrt. Jedoch bleiben einige Nachteile bestehen. Die Modelle müssen für jede Wohnung manuell definiert, trainiert und bei Bedarf, z. B. nach dem Verrücken von Möbelstücken, aktualisiert werden. Die Nutzung von Smart-Home Sensoren erfordert eine relativ aufwendige Installation in der Wohnung des Nutzers. Weiterhin besteht für die Verfahren infolge der Modellierung mittels Farbhistogrammen eine Anfälligkeit für starke Beleuchtungsschwankungen. Zusätzlich kann der Nutzer in sitzender Pose nur an den gelernten und nicht an unbekannten Plätzen detektiert werden. Die Erstellung und Aktualisierung des Nutzermodells generiert einen nicht zu unterschätzenden Aufwand für den Nutzer. Zum Abschluss fehlt die explizite Behandlung von falsch-positiv Detektionen.

5.3. Suche mit Verifikation von Hypothesen

In Volkhardt u. a. (2013c,b) wurde daraufhin ein Verfahren präsentiert, welches die Nachteile aus Volkhardt u. a. (2011a,b) behebt und die Suche von Personen in unterschiedlichen Posen in der gesamten Wohnung ermöglicht. Der Personentracker wird durch rechenaufwendige Detektionsmodule verbessert, die sich nicht im parallelen Betrieb einsetzen lassen und daher bei Bedarf aufgerufen werden. Falsch-positiv Detektionen werden erkannt und anschließend ignoriert. Die Suchstrategie des Roboters wird mit Aufenthaltswahrscheinlichkeitskarten beschleunigt.

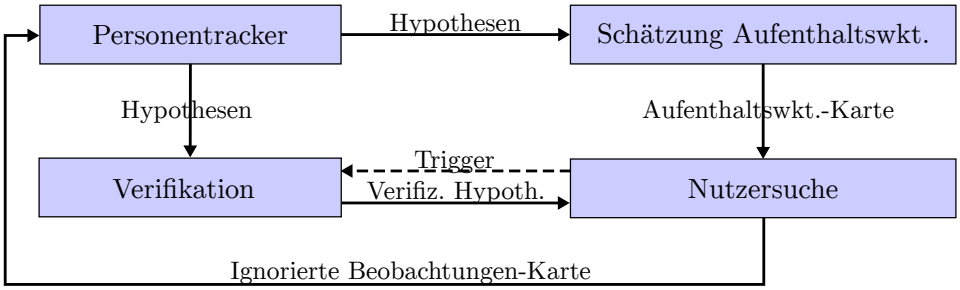


Abbildung 5.3.: Abhängigkeiten der beteiligten Module zur Personensuche.

5.3.1. Suchverhalten und Navigationskonzept

Das Verfahren verwendet den Personentracker aus Kapitel 4 mit den Detektionsmodulen AdaBoost-Gesichtserkennung (Abschnitt 3.3.2), Beinpaardetektion (Abschnitt 3.2.2), HOG Ganzkörper- und Oberkörperdetektion (Abschnitt 3.3.3), um Personen im Erfassungsbereich des Roboters zu tracken.
Verlässt der Nutzer diesen Bereich, wird bei Bedarf das Suchverhalten ausgelöst. Dieses verwendet eine Liste an Navigationspunkten in der Wohnung, welche die Position und Orientierung des Roboters an mehreren Stellen in der Wohnung definieren. Abbildung 5.3 gibt eine Übersicht über die an der Nutzersuche beteiligten Module und deren Abhängigkeiten.
Der Ablauf des Suchverhaltens ist in Algorithmus 5.1 dargestellt. Der Roboter startet die Suche, indem er sukzessive alle Navigationspunkte abfährt; beginnend mit dem Nächstgelegenen. Während der Roboter die Wohnung durchquert überprüft das Suchmodul die Hypothesen des Personentrackers. Falls eine oder mehrere Hypothesen gefunden werden, stoppt der Roboter und dreht sich zur Hypothese mit dem geringsten Abstand und startet den Verifikationsprozess (Abschnitt 5.3.2). Die Verifikation stuft eine Hypothese als sicher oder unsicher ein. Falls die Hypothese nicht verifiziert werden kann, wird sie als Falsch-positive markiert (Abschnitt 5.3.3) und der Roboter fährt mit der Suche fort. Falls eine Hypothese verifiziert wird, nähert sich der Roboter dieser auf Interaktionsdistanz und richtet sein Display nach ihr aus. Anschließend wartet der Roboter einige Sekunden, ob eine Interaktion auf dem Touchscreen erfolgt. Falls ein Buttonklick erfolgt, endet die Suchfahrt und der Nutzer wurde gefunden (Zeile 9), anderenfalls wird die Hypothese als falsch-positiv markiert und die Suche wird fortgesetzt. Diese Heuristik bewirkt, dass der Roboter weiter nach dem Nutzer sucht, auch wenn die Verifikation eine Falsch-positive verifiziert und der Roboter sich dieser nähert. Zusätzlich werden Hypothesen, die der Roboter nicht innerhalb einer vorgegebenen Zeit erreichen kann, als falsch-positiv markiert und ignoriert. Dies kann beispielsweise auftreten, wenn der Weg zur Hypothese versperrt ist. Hat der Roboter alle Navigationspunkte

Algorithmus 5.1 : Ablauf der Personensuche mit Verifikation.

Eingabe : $N = \{\mathbf{n_i}\}$ // Liste von Navigationspunkten

```
1  foreach Navigationspunkt n_i ∈ N do
2      Fahre zu Navigationspunkt                    // Personentracker liefert Hypothesen
3      if Hypothese erkannt then
4          Richtige Kamera zur Hypothese aus        // Rotation des Roboters
5          Starte Verifikation                      // Bewegungs- und DPM-Detektor
6          if Hypothese verifiziert then
7              Nähere dich der Hypothese auf Interaktionsdistanz
8              if Nutzereingabe erkannt then
9                  Beende Personensuche             // Nutzer gefunden
10             else
11                 Markiere Hypothese als falsch-positiv
12         else
13             Markiere Hypothese als falsch-positiv
14 Beende Personensuche                             // Nutzer nicht gefunden
```

besucht, ohne Interaktion des Nutzers, wird die Suche mit dem Signal, dass der Nutzer nicht gefunden wurde, beendet (Zeile 14).

5.3.2. Verifikation von Hypothesen

Die Verifikation wird bei Bedarf immer dann ausgeführt, wenn das Suchmodul eine Hypothese vom Tracker erhält. In Volkhardt u. a. (2013c,b) werden zwei Detektionsmodule eingesetzt, die nicht im parallelen Betrieb eingesetzt werden können.
Zur Verifikation stoppt der Roboter seine Bewegung und triggert die Detektoren. Zunächst wird ein rechenaufwendiges DPM eingesetzt, welches auf dem VOC2009 Datensatz trainiert wurde (P. F. Felzenszwalb u. a. 2010; Everingham u. a. 2009). Wie in Abschnitt 3.3.4 angegeben, konnte durch die C++ Implementierung von Laschka (2013)[2] der Detektor erstmals auf dem mobilen Roboter eingesetzt werden. Die effiziente Variante von Dubout u. a. (2012) war zu diesem Zeitpunkt noch nicht verfügbar. Die Berechnungszeit lag daher bei 2-4 Sekunden pro Bild. Das DPM eignet sich besonders, um sitzende und partiell verdeckte Personen zu erkennen. Als zweites Verfahren wird eine Bewegungsdetektion eingesetzt. Da der Roboter zur Verifikation stillsteht,

[2] Vom Autor im Rahmen dieser Arbeit betreut.

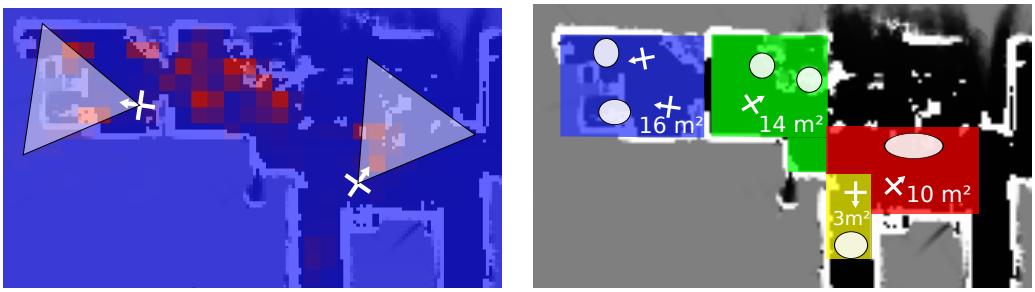
können im Kamerabild Bewegungen mittels Differenzbildern erkannt werden. Erkannte Bewegungen umfassen unter anderem: das nach vorne Lehnen zum Roboter, den Kopf drehen oder das Winken mit einer Hand. Diese Bewegungen entstehen ganz natürlich, wenn Senioren mit dem Roboter interagieren und auf sich aufmerksam machen wollen.
Ohne die Verifikation müsste sich der Roboter jeder Hypothese auf Interaktionsdistanz nähern und anschließend einige Sekunden auf die Eingabe des Nutzers warten. Daher beschleunigt die Verifikation den Suchprozess. Zusätzlich werden die erfolgreichen Suchfahrten erhöht, da der Roboter weniger Gefahr läuft, an Möbeln anzustoßen, während er sich (falsch-positiv) Hypothesen nähert.

5.3.3. Behandlung von Falsch-positiven

Hypothesen, die nicht durch den vorangegangenen Prozess oder durch Nutzerinteraktion verifiziert wurden, werden als Falsch-positive markiert und anschließend ignoriert. Hierfür wird eine rechteckige Region mit einer Breite (2 m) an der Stelle der Hypothese in die Karte der ignorierten Bereiche eingetragen (Abbildung 5.3). Dies bewirkt, dass der Tracker Hypothesen und Detektionen in diesem Bereich ignoriert (Abschnitt 4.9.2) und der Roboter seine Suche in anderen Teilen der Wohnung fortsetzen kann.
Ohne diesen Mechanismus würde der Roboter endlos vor einer, unter Umständen verifizierten, Hypothese warten, beziehungsweise diese immer wieder neu detektieren und anfahren. Um zu verhindern, dass zu viele Regionen in der Wohnung maskiert werden, setzt der Algorithmus die falsch-positiv Markierungen nach einer Suchfahrt zurück.

5.3.4. Aufenthaltswahrscheinlichkeitskarte

Die Auswahl der Navigationspunkte während der Suchfahrt durch die Wohnung kann verbessert werden, indem die Positionen zuerst angefahren werden, welche eine hohe Wahrscheinlichkeit haben, dass sich der Nutzer dort befindet. Daher schätzt ein Modul die Aufenthaltswahrscheinlichkeit des Nutzers in einer Karte, indem die Hypothesen des Personentrackers genutzt werden. Diese Karte wird als metrisches 2D-Histogramm implementiert, wobei jedes Bin einen kleinen Bereich der Wohnung repräsentiert (ähnlich der Zelle einer Occupancy-Map). Jeder Zählwert im jeweiligen Bin signalisiert das Vorhandensein einer Hypothese in diesem Gebiet. Das Histogramm wird nach jeder Aktualisierung normiert, sodass sich eine Aufenthaltswahrscheinlichkeit des Nutzers ergibt.



(a) Aufenthaltswahrscheinlichkeitskarte **(b)** Experimentelle Umgebung

Abbildung 5.4.: (a) Aufenthaltswahrscheinlichkeit des Nutzers in der Testumgebung überlagert mit der Occupancy-Karte. Wahrscheinlichkeit farbcodiert von Blau zu Rot (skaliert für bessere Sichtbarkeit). Weiße Kreuze bezeichnen beispielhafte Navigationspunkte (zuvor festgelegte Roboterposen in der Wohnung); Dreiecke die entsprechenden Sensormessfelder. (b) Occupancy-Karte der Testumgebung mit überlagerten Labels. Blau: Wohnzimmer, grün: Gästezimmer, rot: Küche, Gelb: Flur. Weiße Ellipsen zeigen die Positionen des Nutzers während der Experimente; weiße Kreuze die Navigationspunkte mit entsprechender Orientierung.

Während der Suchfahrt kann der Roboter nicht einfach zur Position mit der höchsten Wahrscheinlichkeit fahren, da diese unter Umständen, z. B. durch verschobene Möbel, nicht erreichbar ist. Weiterhin besteht die Möglichkeit, dass das Sensorsichtfeld an der ausgewählten Position durch Wände blockiert ist und nicht den vollen Bereich der Aufenthaltswahrscheinlichkeitskarte abdeckt. In Abschnitt 5.4 wird ein Verfahren vorgestellt, das den tatsächlichen Sichtbereich des Roboters auf Basis der Hinderniskonfiguration an der Zielposition modelliert und nicht nur die zu beobachtende Aufenthaltswahrscheinlichkeit betrachtet. Im vorliegenden Verfahren von Volkhardt u. a. (2013b) behilft man sich, indem der Algorithmus den Navigationspunkt selektiert, bei dem das Sensormessfeld die größte akkumulierte Wahrscheinlichkeit erfasst. Das Sensormessfeld wird als einfaches Dreieck modelliert, bei dem eine Ecke auf dem Navigationspunkt liegt und die anliegende Winkelhalbierende mit der zugehörigen Orientierung des Roboters zusammenfällt (Abbildung 5.4(a)). Die Länge des Sichtbereichs wird auf 4 m festgelegt und der Öffnungswinkel des Dreiecks auf 30° gesetzt.

Durch Integration der Fläche, die durch das, als Dreieck modelliertes, Sensormessfeld erfasst wird, kann die Aufenthaltswahrscheinlichkeit der Person für jeden Navigationspunkt berechnet werden. Der Roboter wählt den Punkt mit der höchsten Wahrscheinlichkeit, statt den Nächstgelegenen. Sobald der Punkt

besucht und überprüft wurde, wählt der Roboter den Navigationspunkt mit der nächsthöheren Wahrscheinlichkeit, bis alle Punkte abgefahren wurden.
Durch diesen Mechanismus überprüft der Roboter zunächst Gebiete, in denen der Nutzer in der Vergangenheit häufig detektiert wurde. Unter der Voraussetzung, dass sich der Nutzer so verhält wie die gelernte Verteilung, sollte dies die Suchzeit bedeutend verringern. Hält sich der Nutzer an einem ungewöhnlichen Punkt auf, kann es dadurch aber auch vorkommen, dass sich die Suchzeit erhöht.

5.3.5. Experimentelle Ergebnisse

Das Verfahren wurde in einer nachgestellten 3-Raum Testumgebung evaluiert, welche auch für die Evaluation des Personentrackers verwendet wurde (Anhang A.8). Abbildung 5.4(b) zeigt den Grundriss der Räume mit den zugehörigen Navigationspunkten. Die Positionen des Nutzers während der Experimente sind als Ellipsen dargestellt. Der Nutzer saß an jeweils 2 Positionen auf einer Couch im Wohnzimmer, auf 2 Sesseln im Gästezimmer, auf einem Stuhl in der Küche oder stand im Flur.

Verifikation

Zunächst wurde getestet, ob die vorgeschlagene Verifikation von Hypothesen die Fähigkeit des Roboters, Personen in der Wohnung zu finden, verbessert. Diese Experimente nutzten nicht die Aufenthaltswahrscheinlichkeitskarte aus Abschnitt 5.3.4, um die Suchfahrt zu steuern. Es wurden drei Methoden getestet. Die Erste nutzte keine Verifikation, die Zweite nutzte eine Verifikation auf Basis von Bewegung und die Dritte eine Kombination aus DPM und Bewegung. Alle Experimente nutzen die Behandlung von falsch-positiv Detektionen aus Abschnitt 5.3.3.
Für jede Methode startet der Roboter aus 3 unterschiedlichen Startpositionen: Wohnzimmer, Gästezimmer, Küche. Der Nutzer nimmt eine von 6 Positionen in der Wohnung ein (Abbildung 5.4(b)). Für jede Position wurden 3 Suchen durchgeführt. Somit ergeben sich 54 Versuche pro Methode.
Während der Durchläufe wurden die Erfolgsrate und die Zeit zum Finden des Nutzers gemessen. Erfolgreiche Suchen wurden gezählt, wenn der Roboter den Nutzer detektierte, sich diesem näherte und anhielt, sodass der Nutzer den Touchscreen leicht bedienen konnte, um die Suche zu beenden. Fehlversuche wurden gezählt, wenn der Roboter den Nutzer nicht fand, sich diesem nicht näherte, mit einem Hindernis kollidierte oder die Suche länger als 3 Minuten dauerte. Die Ergebnisse der Experimente sind in Tabelle 5.2 zu finden. Die mittleren Zeiten wurden auf erfolgreichen Suchdurchläufen berechnet. Die Er-

Tabelle 5.2.: Erfolgsrate und benötigte Zeit für Suchfahrten unter Nutzung unterschiedlicher Verifikationsmethoden. Die Zeiten beziehen sich auf erfolgreiche Suchfahrten.

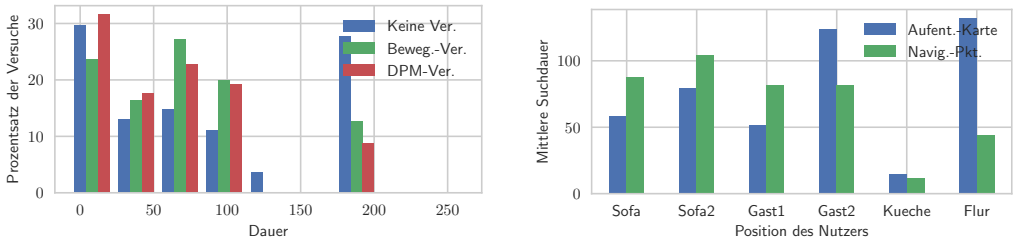
Methode	Erfolgsrate	mittlere Dauer [s]	max. [s]	std [s]
Keine Verif.	0.72	53.1	146	43.9
Bewegungs-Verif.	0.87	59.4	116	34.0
DPM-Verif.	0.91	55.5	115	34.6

folgsrate ohne Verifikation ist bedeutend geringer als bei den Versuchen mit Verifikation. Die DPM-Verifikation erreichte die höchste Erfolgsrate, da der DPM auch stillsitzende Personen verifizierte.
Die häufigsten Gründe für nicht erfolgreiche Suchvorgänge (in absteigender Reihenfolge des Auftretens) waren:

1. Der Roboter stößt an ein Hindernis.
2. Die Suche dauerte länger als 3 Minuten.
3. Der Roboter detektierte den Nutzer nicht.
4. Die Verifikation schlug fehl und maskierte die Hypothese aus.

Das Problem der Hinderniskollision tritt ohne Verifikation gehäuft auf, da der Roboter versucht, sich jeder Hypothese des Personentrackers auf Interaktionsdistanz zu nähern. Dies schließt falsch-positiv Hypothesen auf Tischen, Schränken, in Lücken und engen Ecken des Raumes ein. Trotz Hinderniswahrnehmung kollidiert der Roboter relativ häufig, wenn er sich sehr dicht an Objekten vorbei bewegt oder er sich bei wenig Platz dreht. Das zweite Problem trat ebenfalls häufiger auf, wenn keine Verifikation eingesetzt wurde. Dass der Nutzer nicht detektiert wurde, trat in etwa bei allen Methoden gleich häufig auf. Gründe hierfür war meist eine seitlich sitzende Pose des Nutzers oder verdeckte Beinpaare. Zuletzt kam es in den Experimenten zweimal vor, dass bei der Bewegungs- und DPM-Verifikation die Hypothese des Nutzers nicht verifiziert und als Falsch-positive maskiert wurde.
Die durchschnittliche und maximale Dauer der erfolgreichen Versuche aus Tabelle 5.2 sind relativ groß gegenüber den 80 s, die der Roboter benötigt, um alle Navigationspunkte abzufahren[3]. Wenn keine Verifikation eingesetzt wird,

[3]Der Roboter benötigt 80 s, gemittelt über 5 Durchläufe, um zu jedem der fünf Navigationspunkte der Wohnung zu erfahren während der Personentracker deaktiviert ist. Die Suchdauer beträgt also im Worstcase mindestens diese Zeit.



(a) Eingesetzte Verifikationsmethode **(b)** Eingesetzte Suchstrategie

Abbildung 5.5.: Suchdauer nach Methoden und Strategien. (a) Histogramm der Suchdauer in [s] von unterschiedlichen Verifikationsmethoden. (b) Suchdauer erfolgreicher Suchen mit Suchstrategien auf Basis von Navigationspunkten beziehungsweise einer Aufenthaltswahrscheinlichkeitskarte. Dauer in [s], gemittelt über jeweils 3 Versuche.

erhöht sich die Zeitdauer hauptsächlich durch das Anfahren vieler Hypothesen und das Warten auf eine Nutzereingabe. Bei den Verifikationsmethoden dreht sich der Roboter nur zur Hypothese, startet jedoch die Verifikation, welche eine gewisse Zeit in Anspruch nimmt. Dies erzeugt eine ähnliche durchschnittliche Suchdauer. Für einen besseren Überblick zeigt Abbildung 5.5(a) die Verteilung der Suchzeiten der einzelnen Methoden als Histogramm. Die Bin-Breite beträgt 30 s. Die Werte im letzten Bin enthalten auch nicht erfolgreiche Versuche, welche länger als 3 Minuten (180 s) benötigten. Wenn keine Verifikation eingesetzt wird, findet der Roboter den Nutzer entweder sehr schnell (Bin 1) oder erst sehr spät beziehungsweise gar nicht. Die Bewegungs- und DPM-Verifikation bewirken viele kurze und mittlere Suchdauern mit wenigen langen Suchdauern.

Aufenthaltswahrscheinlichkeitskarte

Für die Generierung der Karte, wurde der Nutzer bei kurzen täglichen Aktivitäten und während der Experimente aus Abschnitt 5.3.5 getrackt und die Aufenthaltswahrscheinlichkeit geschätzt. Die resultierende Karte ist in Abbildung 5.4(a) abgebildet. Die höchste Wahrscheinlichkeit den Nutzer anzutreffen, liegt im Wohnzimmer, gefolgt von Gästezimmer, Küche und Flur. Diese gewonnene Wahrscheinlichkeitsverteilung ist dabei stark abhängig von den zuvor durchgeführten Experimenten und ist nicht als allgemeingültig anzusehen. Im Anwendungsszenario muss die Karte längerfristig unter realem Verhalten des Nutzers erstellt und aktualisiert werden.
Um die Unterschiede zwischen den beiden Suchstrategien aufzudecken, wurde

die Startposition des Roboters fest auf die Küche gesetzt und die Experimente aus Abschnitt 5.3.5 wiederholt. Abbildung 5.5(b) zeigt die durchschnittliche Suchdauer, um den Nutzer auf einer der sechs Positionen zu finden – gemittelt über jeweils 3 Versuche. Wenn die Aufenthaltswahrscheinlichkeitskarte genutzt wird, ist die Suchdauer für beide Sofapositionen und die Gast1 Position kürzer als unter Verwendung der Navigationspunkte. Der Roboter prüft im ersten Fall nicht die nahegelegenen Positionen des Flurs und des Gästezimmers, sondern fährt direkt ins Wohnzimmer. Auf dem Weg dorthin fällt die Position Gast1 in den Sensorsichtbereich, was ebenfalls zu einer kürzeren Suchdauer führt. Ein Nachteil der kartenbasierten Suche fällt bei den Gast2 und Flur Positionen auf. Der Roboter fährt zunächst durch das Gästezimmer in das Wohnzimmer und ignoriert die nahegelegene Gast2 und Flur Position. Diese werden erst in der Reihenfolge der Aufenthaltswahrscheinlichkeit des Nutzers später überprüft, was in einer höheren durchschnittlichen Suchzeit resultiert. Im Falle der Küche wird der Nutzer bei beiden Strategien direkt gefunden.

5.3.6. Zusammenfassung

In diesem Abschnitt wurden Strategien und Methoden vorgestellt, die es einem mobilen Roboter ermöglichen, Personen in häuslichen Umgebungen zu suchen und sich diesen auf Interaktionsdistanz zu nähern. Während der Suchfahrt durch die Wohnung nutzt der Roboter bei Bedarf einen Bewegungs- und DPM-Detektor, um Hypothesen des Personentrackers zu verifizieren und Falsch-positive auszuschließen (Volkhardt u. a. 2013c). Weiterhin wurden zwei Suchstrategien auf Basis von Navigationspunkten und einer Aufenthaltswahrscheinlichkeitskarte vorgestellt, die die Route des Roboters durch die Wohnung festlegen (Volkhardt u. a. 2013b).

Über 150 experimentelle Suchfahrten in einer nachgestellten Wohnung konnten zeigen, dass die Verifikation die Erfolgsrate der Personensuche verbessert und die maximale Suchzeit verringert wird. Durch Nutzung der Aufenthaltswahrscheinlichkeitskarte kann die durchschnittliche Suchzeit verringert werden, falls sich der Nutzer nach dem gelernten, auf ihn personalisierten Modell verhält.

In weiterführenden Arbeiten könnte die Karte zusätzlich Kontextinformationen erfassen. Beispielsweise könnte die Wahrscheinlichkeit, dass sich die Person gegen Mittag in der Küche und abends auf dem Sofa befindet, erhöht werden. Im folgenden Abschnitt wird eine verbesserte explorative Suchstrategie beschrieben, welche die Nutzung von festen Navigationspunkten überflüssig macht und die Aufenthaltswahrscheinlichkeit auf natürliche Weise integriert.

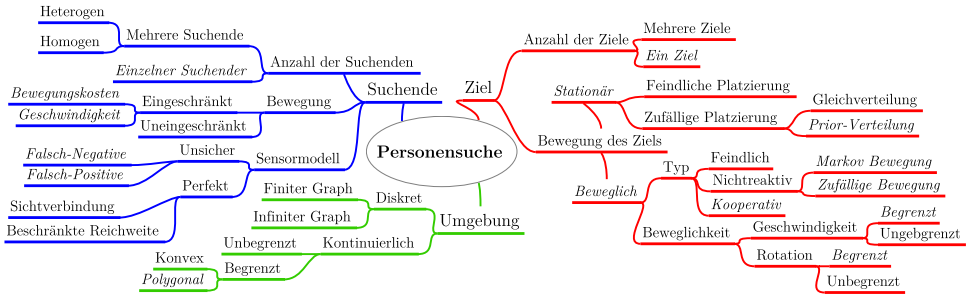


Abbildung 5.6.: Aspekte explorativer Suchalgorithmen zur Personensuche. Kursiv markierte Parameter beschreiben das Szenario der Personensuche dieser Arbeit (Vgl. Chung u. a. (2011)).

5.4. Explorative Suche

In diesem Abschnitt wird eine Suchstrategie vorgestellt, bei der der Roboter explorativ an Stellen sucht, an denen der Nutzer vermutet wird, beziehungsweise bisher noch nicht gesucht wurde.

5.4.1. Einordnung in den wissenschaftlichen Kontext

In Verbindung mit mobilen Robotern treten explorative Verfahren häufig im Kontext von Kartenaufbau (Amigoni u. a. 2010; Vallvé u. a. 2015), aktiver Lokalisierung (Thrun u. a. 2005) beziehungsweise SLAM auf (Holz u. a. 2010; Valencia u. a. 2012). Hierbei maximiert der Roboter den erwarteten Informationsgewinn über die Karte beziehungsweise seine Position. Eine weitere verwandte Problemstellung umfasst die Pursuit-Evasion Probleme (Borie u. a. 2011), bei denen eine Verfolgergruppe versucht, eine andere Gruppe der Verfolgten aufzuspüren und zu fangen. Eine Übersicht relevanter automatisierter Suchverfahren für mobile Roboter in Pursuit-Evasion Szenarien findet sich in Chung u. a. (2011). In Abbildung 5.6 sind verschiedene Aspekte dieser Suchszenarien aufgelistet. Kursiv markiert sind die Parameter, welche in der vorliegenden Arbeit eine Rolle spielen.
Ein ähnlicher Ansatz zum vorgestellten Algorithmus zur explorativen Suche von Personen ist in (Stückler u. a. 2011) zu finden. Dieser nutzt eine semantische Priorverteilung in Form einer Präsenzkarte, um den Roboter bei der Exploration an Stellen zu senden, bei denen Personen mit hoher Wahrscheinlichkeit anzutreffen sind (Stühle, begehbarer Untergrund) und Falsch-positive auszuschließen (in Wänden). Die Verifikation von Hypothesen erfolgt aus-

schließlich durch Gesichtsdetektionen (Viola u. a. 2002). Die Aktualisierung der Priorkarte erfolgt mithilfe des Occupancy Grid Mapping Update Algorithmus (Thrun u. a. 2005) in Bereichen um verifizierte Hypothesen. Zur Exploration werden normalverteilte Posen um die aktuelle Roboterposition gesampelt und die Zielpose ausgewählt, welche die Detektionswahrscheinlichkeit einer Person in der Präsenzkarte maximiert. Bereits durchsuchte Regionen mit gefundenen Personen werden markiert und anschließend nicht erneut aufgesucht.
Bei allen vorgestellten Verfahren kommt ein Greedy Algorithmus zur Anwendung, der nur die nächste Zielposition plant (*next-best-view*) und den unbekannten Informationsgewinn an der Zielposition approximiert. Weiterhin wird häufig der Informationsgewinn durch Beobachtungen des Roboters auf dem Weg zur Zielposition ignoriert. Der folgende Abschnitt beschreibt, die in dieser Arbeit entwickelte explorative Suchstrategie.

5.4.2. Explorationsstrategie zur Nutzersuche

Der prinzipielle Ablauf der Suche ist weitgehend identisch mit dem Verfahren von Volkhardt u. a. (2013b) aus Abschnitt 5.3. Die Suchstrategie ist jedoch *explorativ* und basiert nicht auf festgelegten Navigationspunkten. Weiterhin wurden einige Details im Verfahren verbessert.
Zur Auswahl von neuen Zielpunkten wird beim verwandten explorativen Grid-Mapping häufig die Entropie $H_p(x) = -\int p(x) \log p(x)\, dx$ als erwartete Information $E[-\log p]$ der Occupancy-Karte verwendet (Thrun u. a. 2005). Dabei haben unbeobachtete Zellen die größte Unsicherheit (Entropie) und damit auch den größtmöglichen erwarteten Informationsgewinn. Sicher belegte oder freie Zellen haben im Gegenzug eine geringe Entropy mit geringem Informationsgewinn.
Das vorliegende Verfahren nutzt eine Personenwahrscheinlichkeitskarte, die der Aufenthaltswahrscheinlichkeitskarte aus Abschnitt 5.3.4 ähnelt. Im Gegensatz zum explorativen Grid-Mapping soll der Roboter nicht bevorzugt unbekannte Zellen beobachten, sondern vor allem Zellen mit hoher Wahrscheinlichkeit einer Person. Durch die Dynamik der Person ist es möglich, dass beobachtete Zellen mit geringer Wahrscheinlichkeit später eine hohe Personenwahrscheinlichkeit erhalten, falls sich der Nutzer in diese Zelle bewegt. Für die Aktualisierung der Gridzellen und zum Erfassen der Personenwahrscheinlichkeit an einer Position muss ein Sensormodell definiert werden.

Sensormodell

Der Sensorsichtbereich des Roboters wird nicht als einfaches Dreieck wie in Abschnitt 5.3.4 sondern als virtueller Scanner, ähnlich einem Lasersensor, de-

finiert. Der Sichtradius beträgt 2 m, um eine robuste Erkennung durch die visuellen Detektoren zu ermöglichen; der Öffnungswinkel beträgt 60°. Der Sensorkegel wird in 10 virtuelle Scans mit jeweils 6° Öffnungswinkel eingeteilt. Beispielhafte Scans sind in Abbildung 5.7 zu finden. Die virtuellen Scans dringen nicht durch belegte Zellen einer Occupancy-Karte. Diese enthält Hindernisse, die den Sichtbereich des Roboters blockieren, beispielsweise Wände, aber keine niedrigen Tische (vgl. Abschnitt 4.9.2). Das Update der Zellen und die Ermittlung der Wahrscheinlichkeit in der Personenwahrscheinlichkeitskarte erfolgt nur in den freien Gridzellen dieser Occupancy-Karte.

Aktualisierung der Personenwahrscheinlichkeit Während sich der Roboter durch die Wohnung bewegt, wird die Personenwahrscheinlichkeitskarte im Bereich des Sensorsichtfeldes aktualisiert. Analog zum Prinzip des Occupancy-Mapping Algorithmus (Thrun u. a. 2005) wird für jede Zelle $\mathbf{m}_i$ im Sensorkegel des Roboters der Personenwahrscheinlichkeitskarte ein binärer Bayes-Filter (Abschnitt 4.4) in *log odds* Form[4] verwendet:

$$l_{t,i} = l_{t-1,i} + log\frac{p(\mathbf{m}_i|z_t, x_t)}{1 - p(\mathbf{m}_i|z_t, x_t)} - l_0 \tag{5.3}$$

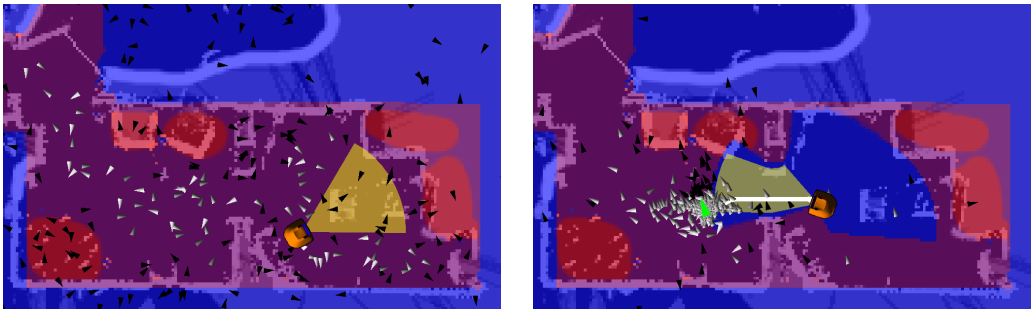
wobei $l_{t,i}$ dem *log odds* Verhältnis von $p(\mathbf{m}_i|z_{1:t})$ entspricht. l_0 stellt den Prior jeder Zelle in *log odds* Form bevor Sensormessungen getätigt werden dar. Der Belief von Zellen, die sich nicht im Updatebereich des Sensorkegels liegen, wird nicht verändert. Eine Beschreibung des Algorithmus findet sich in Anhang A.11.2.

Erfassung der Personenwahrscheinlichkeit Mithilfe des Sensormodells ist es möglich, die Güte jeder Zielposition in der Karte zu bestimmen. Hierfür wird die Personenwahrscheinlichkeit aller Zellen, die sich im Updatebereich des Sensorkegels befinden, akkumuliert.

Partikelschwarm Optimierung

Der Suchraum der möglichen Zielpositionen ist verhältnismäßig groß und umfasst alle erfahrbaren Positionen und Orientierungen. Zur Abdeckung des Suchraums bieten sich daher Sampling-Verfahren an. In dieser Arbeit wurde eine Partikelschwarm Optimierung (PSO) verwendet (Kennedy u. a. 1995). Eine Beschreibung des Algorithmus findet sich in Anhang A.11.1.

[4] Die *log odds* berechnen den Logarithmus einer Wahrscheinlichkeit in *odds* Form: $\text{logit}(p) = \log(\frac{p}{1-p})$



(a) Initiale Partikelverteilung **(b)** Konvergierte Partikel

Abbildung 5.7.: Roboter in Testumgebung mit überlagerter Personenwahrscheinlichkeit (farbcodiert Blau zu Rot). Virtueller Sichtbereich in gelb. Partikelposen mit Orientierung als Dreiecke dargestellt. Kosten farblich von Schwarz (hoch) zu Weiß (gering) codiert. (a) Initial werden die Partikel gleichverteilt eingestreut. Partikel im Freiraum haben geringe Kosten. Partikel in der Nähe von Hindernissen haben hohe Kosten. (b) Konvergierte Partikelverteilung mit neuer Zielpose im Partikelschwarm Optimum (Grün), sowie geplanter Pfad (Weiß). Bereits explorierte Gebiete ohne Personen erhalten geringe Wahrscheinlichkeit (Blau).

Zu Beginn der Optimierung werden alle Partikel gleichverteilt in Position und Orientierung im Freiraum der Wahrscheinlichkeitskarte eingestreut (Abbildung 5.7(a)). Nach der PSO wählt der Roboter das globale Partikelschwarm Optimum als nächste Zielposition (Abbildung 5.7(b)). Die Kostenfunktion für die Bewertung der Partikel unterscheidet sich, je nachdem ob eine Hypothese vom Tracker erkannt wurde (Annäherung) oder nicht (Exploration).

Kostenfunktion für Exploration Liefert der Tracker keine Hypothese, soll eine Zielposition ausgewählt werden, die eine hohe Wahrscheinlichkeit besitzt, eine Person zu finden und die schnell und sicher erreicht werden kann. Für jedes Partikel p_i wird folgende Kostenfunktion verwendet:

$$s = -p(x_i)\ f_p + c_h \tag{5.4}$$

Die einzelnen Terme der Funktion sind in Abbildung 5.8 noch einmal bildlich veranschaulicht. In der Kostenfunktion bezeichnet $p(x_i)$ die Personenwahrscheinlichkeit, f_p den Pfadkostenfaktor und c_h die Hinderniskosten an der Zielposition.
Die Personenwahrscheinlichkeit $p(x_i)$ berechnet sich als akkumulierte Summe

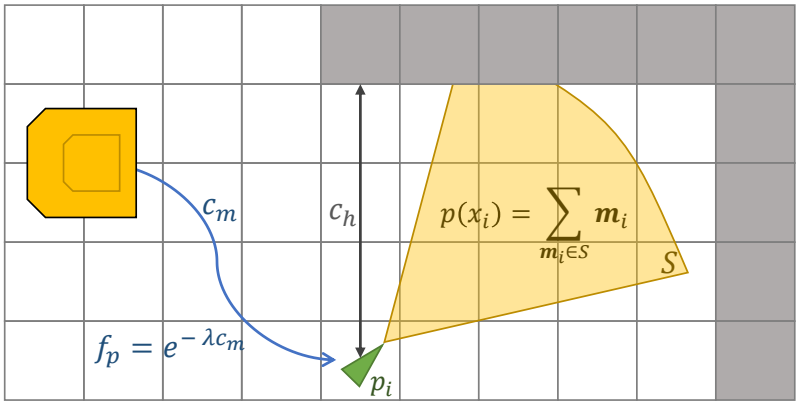


Abbildung 5.8.: Kostenfunktionen für die Bewertung eines Partikels bei Exploration. Roboter in Belegtheitskarte mit belegten Gridzellen in grau. Die Kosten eines Partikels p_i ergeben sich aus der negativen zu erwartenden Personenwahrscheinlichkeit -$p(x_i)$, dem Pfadkostenfaktor f_p, um die Partikelposition zu erreichen und den Hinderniskosten c_h, die den Abstand des Partikels zu Hindernissen bewerten.

über alle Zellen $\mathbf{m}_i$ im Sensorkegel S der jeweiligen Position des Partikels p_i:

$$p(x_i) = \sum_{\mathbf{m}_i \in S} \mathbf{m}_i \tag{5.5}$$

Im Zuge der Optimierung wird nur die erwartete Personenwahrscheinlichkeit des Partikels (Zielpunkt) in Betracht gezogen, nicht aber mögliche Beobachtungen, die der Roboter auf dem Weg zur Zielposition tätigt[5].
Der Pfadkostenfaktor f_p aus Gleichung (5.4) bewertet die Distanz des Partikels zum Roboter. Er sinkt mit zunehmender Distanz und wird mittels abfallender Exponentialfunktion aus den Pfadkosten c_m bestimmt:

$$f_p = \mathrm{e}^{-\lambda c_m} \tag{5.6}$$

Die Pfadkosten c_m ergeben sich aus einer Kostenkarte, welche die Wegkosten zu jeder befahrbaren Zelle der Gridkarte enthält. Eine effiziente Berechnung erfolgt mittels E-Stern Algorithmus (Philippsen u. a. 2005) ausgehend vom

[5]Ein Sampling der Trajektorie zum Zielpunkt und die Akkumulation der Personenwahrscheinlichkeiten ist mit hohem Berechnungsaufwand pro Partikel verbunden. Zusätzlich entspricht der geplante Pfad selten dem tatsächlich gefahrenen (unsichere Regler, dynamische Hindernisse)

Roboter[6]. Die Orientierung wird dabei vernachlässigt. Partikelpositionen, die vom Roboter nicht erreichbar sind, erhalten hierdurch maximale Kosten. Effektiv reduziert der Pfadkostenfaktor mit zunehmender Distanz die akkumulierte Personenwahrscheinlichkeit $p(x_i)$. Der Diskontierungsfaktor λ bestimmt, wie schnell der Faktor gegen 0 läuft. Ein kleines λ führt dazu, dass der Roboter große Explorationswege vollführen kann, um eine Position mit hoher Wahrscheinlichkeit zu erreichen. Als Resultat sucht der Roboter zunächst die gesamte Umgebung grob ab und füllt erst am Ende verpasste Lücken. Ein großes λ bewirkt, dass der Roboter beständig alle Bereiche erfasst und kurze Wege bevorzugt.
Die Hinderniskosten c_h an der Zielposition aus Gleichung (5.4) ergeben sich aus einer distanztransformierten, mit dem Robotergrundrissradius dilatierten Occupancy-Karte (P. Felzenszwalb u. a. 2012). Die Kosten bestrafen Partikel, welche nahe an Hindernissen liegen. Befindet sich das Partikel in einem Hindernis oder würde der Roboter an dieser Position mit dem Hindernis kollidieren, werden maximale Kosten angenommen (Abbildung 5.7(b)).

Kostenfunktion für Annäherung an Nutzer Sobald der Personentracker eine Hypothese liefert, ist das Ziel des Roboters eine möglichst gute Interaktionsposition einzunehmen. Die zugehörige PSO wird sofort gestartet, auch wenn sich der Roboter gerade auf dem Weg zu einer Explorationsposition befindet. Dabei werden die Partikel normalverteilt um die Hypothese eingestreut.
Die Kosten für ein Partikel ergeben sich zu:

$$s = c_d + c_r + w_h c_h + w_m(1 - \mathrm{e}^{-c_m}) \tag{5.7}$$

Hierbei bestimmt c_d die Kosten für die Distanz zum optimalen Interaktionsabstand des Roboters zum Nutzer und c_r die Rotationskosten für die Winkelabweichung zur gewünschten Blickrichtung (Display zum Nutzer gewandt). Die Hinderniskosten c_h und die Pfadkosten c_m ergeben sich analog zu Gleichung (5.4), werden jedoch mit w_h und w_m gewichtet. Der Faktor $w_h = 0.1$ ermöglicht dem Roboter, möglichst nah an Hindernisse heranzufahren. Während bei der Exploration Positionen im Freiraum bevorzugt werden, zu denen sich der Roboter schnell bewegen kann, ist es bei der Interaktion meist nötig, dass sich der Roboter nah an Hindernisse (z. B. Stühle) heranbewegt. Die Pfadkosten c_m werden mithilfe des Terms $1 - \mathrm{e}^{-c_m}$ auf den Bereich $[0 \ldots 1]$, bzw. mit $w_m = 0.5$ auf $[0 \ldots 0.5]$ normiert. Dies bewirkt, dass der Roboter zwar kurze Wege bevorzugt, eine geringe Distanz zum Nutzer aber Vorrang besitzt (Abbildung 5.9).

[6] Als einfache Approximation kann, bei nicht Vorhandenseins eines Pfadplanungsalgorithmus, auch der euklidische Abstand verwendet werden.

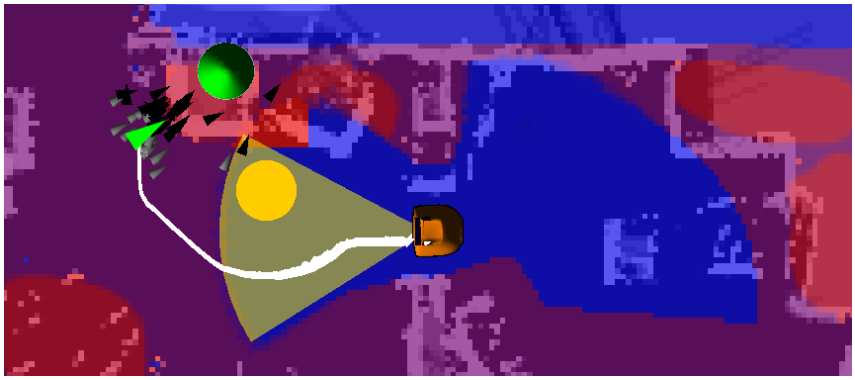


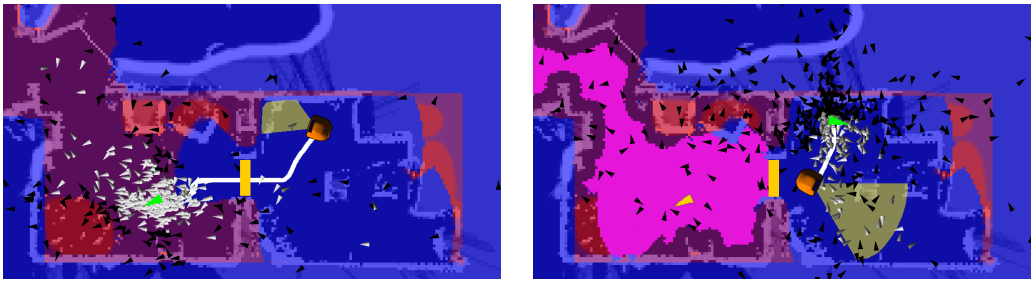
Abbildung 5.9.: Annähern an Nutzer. Erkannte Nutzerhypothese (grüner Kreis). Konvergierte Partikelverteilung mit Positionen in Interaktionsdistanz und Displayausrichtung zur Person (Dreiecke). Beste Zielposition als grünes Dreieck dargestellt. Der Roboter fährt einen Bogen um einen Tisch (orange markiert), um sich möglichst nahe am Nutzer zu platzieren. Sonstige Elemente analog zu Abbildung 5.7(b).

Besonderheiten beim Annähern an den Nutzer

Während der Anfahrt zur besten Interaktionsposition wird die Nutzerhypothese aus dem Personentracker fortlaufend auf Positionsänderungen überprüft. Überschreitet die Positionsänderung eine festgelegte Schwelle von 50 cm, wird eine erneute PSO durchgeführt, um zu einem neuen Zielpunkt zu planen, welcher näher am Nutzer liegt. Positionsänderung von Hypothesen können einerseits durch die Dynamik des Nutzers verursacht werden; zum Beispiel, wenn sich dieser auf den Roboter zubewegt oder der Nutzer sich und den Roboter während der Anfahrt an eine andere Stelle bewegen möchte (eine Art Folgeverhalten). Andererseits können aus der Ferne beobachtete Hypothesen, vor allem ausschließlich visuell detektierte, beim Näherkommen durch zusätzliche Sensormessungen in ihrer Position durch den Tracker korrigiert werden.
Reagiert der Nutzer nach dem Annähern nicht via Interaktion mit dem Roboter, weil eine falsch-positiv Detektion vorliegt, wird neben den Maßnahmen aus Abschnitt 5.3.3 zusätzlich die Personenwahrscheinlichkeit im Bereich um die Hypothese verringert, um den Roboter in anderen Gebieten nach dem Nutzer suchen zu lassen.

Dynamische Hindernisse

Ein gewählter Zielpunkt ist nach Planung unter dem aktuellen Wissen des Roboters stets im Freiraum. Während sich der Roboter zum Ziel bewegt, kann es vorkommen, dass dieses durch neue Sensoreindrücke plötzlich als ein Hindernis



(a) Planung in unerreichbares Gebiet **(b)** Unerreichbares Gebiet wird ausgeschlossen

Abbildung 5.10.: Umgang mit dynamischen Hindernissen. Elemente analog zu Abbildung 5.7(b). (a) Roboter plant in ein unerreichbares Gebiet (grüne Zielposition), da sich das dynamische Hindernis (orange) nicht im Sensorsichtbereich befindet. (b) Ausgehend vom vorherigen Zielpunkt (oranges Dreieck) wird das nicht erreichbare Gebiet mit der geringsten Wahrscheinlichkeit gefüllt (Rosa) und in zukünftigen Explorationen ignoriert. Neue erreichbare Zielposition in grün.

wahrgenommen wird. Ein Beispiel hierfür sind verschobene Möbel. In diesem Fall sendet das Navigationsmodul des Roboters ein Signal, dass das Ziel nicht erreichbar ist und das Suchmodul kann eine erneute PSO starten, welche mit dem neu erlangten Wissen direkt einen anderen Zielpunkt auswählt. Dies trifft sowohl für die Exploration als auch für das Annähern an eine Person zu.
Weiterhin kann während der Fahrt zum Ziel der Fall eintreten, dass der Pfad zur Zielposition blockiert ist, zum Beispiel durch geschlossene Türen (Abbildung 5.10(a)). Hier erfolgt ebenfalls ein Signal des Navigationsmoduls. Nachdem der Roboter einen anderen Explorationspunkt ausgewählt hat, würde dieser aber unter Umständen erneut versuchen, in den verschlossenen Raum zu fahren. Grund hierfür ist, dass lokale Hindernisse, wie geschlossene Türen, relativ schnell in der Occupancy-Karte gelöscht werden[7]. Um dies zu verhindern, werden bei ankommendem Signal des Navigationsmoduls, dass das Ziel nicht erreichbar ist, die Personenwahrscheinlichkeitswerte der nicht erreichbaren Bereiche auf einen Minimalwert herabgesetzt. In Abbildung 5.10(b) ist dies bildlich dargestellt. Die Ermittlung der Bereiche erfolgt mittels Pfadplanungsalgorithmus ausgehend vom nicht erreichbaren Zielpunkt (oranges Dreieck) und erfasst alle Gridzellen, die hinter dem Hindernis liegen und für den Roboter nicht erreichbar sind (rosa Bereich). Diese Prozedur bewirkt, dass

[7] Andernfalls würden durch dynamische Hindernisse belegte Zellen bei Nichtbeobachtung zunehmend den Bewegungsraum des Roboters einschränken

Tabelle 5.3.: Ergebnisse verschiedener Suchstrategien.

Suchstrategie	Suchfahrten	Erfolgsrate	durchschn. Zeit
Navigationspunkte	134	0.71	47 s
Explorativ	123	0.76	51 s

der Algorithmus für Räume, die hinter einer verschlossenen Tür liegen, eine so niedrige Personenwahrscheinlichkeit annimmt, dass Partikel keinen hohen Score erreichen können und in der PSO nicht ausgewählt werden. Nach einer festgelegten Dauer oder bei Beendigung der Suche werden die abgesenkten Werte zurückgesetzt, um dem Roboter eine spätere Exploration in diese Gebiete zu ermöglichen.

5.4.3. Experimentelle Untersuchungen

Laborumgebung

Die Evaluation wurde in der selben Testumgebung wie in Abschnitt 5.3.5 durchgeführt (Anhang A.8). Allerdings wurde die Testumgebung mit zusätzlichen Einrichtungsgegenständen (Pflanzen, Stehlampen) und Deko (Bilder, Kerzenständer) ausgestattet. Als zusätzlicher Unterschied durften sich acht verschiedene Testnutzer an jeder beliebigen Position der Wohnung, in einer beliebigen Pose, aufhalten. Die Einschränkung der Nutzerpositionen auf festgelegte Punkte (Abbildung 5.4(b)) konnte nun entfallen. Ähnlich zu den Experimenten in Abschnitt 5.3.5 wurden Erfolgsrate und Dauer gemessen.
Zum besseren Vergleich wurden unter gleichen Bedingungen Experimente mit 12 Testpersonen mit dem Verfahren aus Abschnitt 5.3 durchgeführt. Aufgrund einer verbesserten Implementierung konnte das DPM Modul im Dauerbetrieb (siehe Abschnitt 3.3.4) als Detektor für beide Varianten eingesetzt werden. Daher wird auch für das Verfahren aus Abschnitt 5.3 keine Verifikation verwendet. Die Verfahren unterscheiden sich demnach nur in ihrer Suchstrategie: explorativ beziehungsweise festgelegte Navigationspunkte.
Tabelle 5.3 zeigt die Ergebnisse der Experimente. Die Erfolgsrate der explorativen Suche liegt mit 76 % über der Erfolgsrate der Suche mit festen Navigationspunkten (71 %). Grund hierfür ist die freie Wahl der Aufenthaltsposition während der Suchfahrten. Die Testnutzer durften sich auf Stühle, deren Position verändert wurde, setzen beziehungsweise sich an jeder beliebigen Position in der Wohnung aufhalten. Durch die explorative Suchstrategie wird die gesamte Wohnfläche durchsucht und der Nutzer auch an ungewöhnlichen Stellen gefunden. Im Gegensatz dazu kann der Nutzer mit der Suchstrategie auf Basis von Navigationspunkten nur gefunden werden, wenn er sich an den festgeleg-

Tabelle 5.4.: Ergebnisse der Nutzersuche in realen Wohnungen (Gross u. a. 2015).

Umgebung	Anzahl	Erfolgsrate	Fahrstrecke d (m) Mittel	Fahrstrecke d (m) max	Zeit (min) Mittel	Zeit (min) max	$\overline{v}$ (m/s)	Umwegfaktor Mittel	Umwegfaktor max
WhgM1	15	0.73	7.52	32.65	0:54	3:53	0.10	1.66	3.63
WhgM2	18	0.89	13.75	34.30	1:30	3:30	0.12	2.36	9.27
WhgM3	21	0.76	11.22	28.19	1:10	2:43	0.13	1.65	2.91
WhgS1	6	1.00	1.60	3.80	0:13	0:25	0.13	1.05	1.18
WhgS2	13	1.00	1.09	5.15	0:21	1:10	0.04	1.11	1.25
WhgS3	11	0.91	2.82	7.63	0:24	1:02	0.06	1.01	1.03
WhgS4	14	0.93	4.88	8.05	0:39	0:59	0.10	1.10	1.22

ten Stellen befindet, bzw. während der Fahrt von einem Punkt zum anderen durch den Roboter gefunden wird. Die Erfolgsrate ist damit stark abhängig von der manuellen Wahl der Navigationspunkte. Allgemein ist die Erfolgsrate der explorationsbasierten Suche gegenüber den früheren Experimenten aus Abschnitt 5.3.5 niedriger, da die Testbedingungen durch eine zusätzliche Wohnungseinrichtung und die freie Platzwahl erschwert und realistischer wurden (vgl. auch Tabelle A.2).
Im Schnitt benötigt der Roboter bei der explorativen Suchstrategie etwas länger, um den Nutzer zu finden. Dies ist leicht nachvollziehbar, da der Roboter nicht nur fünf festgelegte Zielpunkte anfährt, sondern viele Beobachtungsposition einnimmt, um den gesamten Wohnungsbereich abzudecken.
Zuletzt wurde die mittels PSO ermittelte nahe Interaktionsposition von den Testnutzern als sehr viel angenehmer empfunden, als das einfache gerade Anfahren der navigationsbasierten Suchstrategie.

Reale Wohnumgebung

Im Zuge des SERROGA Projekts (SERROGA 2012) wurde das Suchverhalten des Roboters in Funktions- und Nutzertests mit 3 Mitarbeitern des Fachgebiets NI&KR und 4 Senioren in deren Wohnungen evaluiert (Gross u. a. 2015). Im Gegensatz zu den vorangegangenen Experimenten und den Funktionstests in den Mitarbeiterwohnungen (zufällige Start- und Nutzerposition) wurde das Suchverhalten in den Seniorenwohnungen während der Nutzertests evaluiert. Das heißt die Nutzer interagierten normal mit dem Roboter und eine Suche wurde beispielsweise durch Terminerinnerungen, Videoanrufe oder per Fernbedienung ausgelöst. Die Auswertung der Suchfahrten wurde im Nachhinein durchgeführt. Die Grundrisse der Wohnungen sind in Anhang A.8 zu finden.
Die Ergebnisse sind in Tabelle 5.4 zusammengefasst. Für die Tests in jeder

Umgebung sind jeweils die Anzahl der Suchfahrten, die Erfolgsrate und die benötigte Zeit über die erfolgreichen Fahrten dargestellt. Weiterhin ist jeweils die durchschnittliche und maximale gefahrene Strecke, die durchschnittliche Geschwindigkeit $\overline{v}$ sowie das Verhältnis zwischen gefahrener und minimal notwendiger Strecke dargestellt. In 75% der 54 Suchfahrten in den Wohnungen der Projektmitarbeiter (WhgM1-M3) war die Suche erfolgreich, und der Roboter fand den Nutzer im Mittel in unter 1:30 min. Die längste Suche benötigte 3:53 min. In der Tabelle sind ebenfalls die kürzeste Fahrstrecke d von der Roboterstartposition und dem Nutzer sowie der Umwegfaktor angegeben. Dieser beträgt 1, wenn der Roboter direkt auf dem kürzesten Weg zum Nutzer fährt, und steigt proportional zum zusätzlich zurückgelegten Weg; beispielsweise beträgt er bei doppelter Wegstrecke 2. In den geräumigen Mitarbeiterwohnungen legte der Roboter im Mittel 60% längere Strecken als die kürzeste Entfernung zum Nutzer zurück[8]. In den räumlich kleineren Seniorenwohnungen (WhgS1-S4) konnte der Roboter den Nutzer durch kürzere Suchfahrten finden, beziehungsweise ihn bereits von seiner Ruheposition aus erkennen. Daher wurden die hohe Erfolgsrate von 95% aus 44 Durchläufen, eine geringe Suchzeit und ein geringer Umwegfaktor erzielt. Weiterhin ist ersichtlich, dass in den Tests bei den Senioren S2 und S3 die durchschnittliche Geschwindigkeit deutlich geringer ist, als in den anderen Versuchen. Dies ist darauf zurückzuführen, dass hier viele Suchläufe über sehr kurze Strecken ausgelöst wurden, die für die Anfahrt zum Nutzer eher Rotationsbewegungen und Feinjustierungen auf engem Raum als schnelle Vorwärtsfahrt benötigten.

5.4.4. Zusammenfassung

In diesem Abschnitt wurde eine explorative Suchstrategie vorgestellt, die es dem Roboter ermöglicht, den gesamten Bereich der Wohnung nach Personen abzusuchen. Dabei kann im Gegensatz zum Verfahren aus Abschnitt 5.3 auf manuell festgelegte Navigationspunkte verzichtet werden. Zur Auswahl des nächsten Explorationspunktes wird ein, mittels Partikelschwarm Optimierung ermittelter, Kompromiss aus geringer Wegstrecke, hohem Informationsgewinn und geringer Kollisionswahrscheinlichkeit gewählt. Dynamische Hindernisse, wie verrückte Stühle und geschlossene Türen, werden behandelt. Über Priorwissen, z. B. andere Sensoren oder vergangene Beobachtungen, kann der Roboter bestimmte Regionen der Wohnung bei der Suche bevorzugen. Die Partikelschwarm Optimierung wird ebenfalls genutzt, um eine möglichst nahegelegene Position zum Nutzer anzufahren, bei der das Display gut bedienbar ist. Expe-

[8] Der Umwegfaktor ist in der Regel größer als 1, da der Roboter die tatsächliche Position des Nutzers nicht kennt und somit zusätzliche Bereiche erfahren muss.

rimentelle Untersuchungen bestätigen die Verbesserung in der Erfolgsrate der Suche gegenüber dem Navigationskonzept mit festen Zielpunkten. Neben Versuchen unter nachgestellten Wohnungsbedingungen im Labor kam das Verfahren erfolgreich in mehrtägigen Nutzertests in Seniorenwohnungen zum Einsatz.

5.5. Diskussion und Fazit

In diesem Kapitel wurden drei Verfahren beschrieben, um Personen in häuslichen Umgebungen zu finden, wenn sich diese außerhalb des aktuellen Sichtbereiches des Roboters befinden. Die historisch entstandenen Verfahren bauen aufeinander auf und beseitigen jeweils die Probleme des Vorgängerverfahrens. Das Verfahren aus Abschnitt 5.2 benötigt vorab trainierte Hintergrundmodelle von leeren Plätzen der Wohnung und des Nutzers. Es erfordert eine statische Umgebung und ist anfällig gegenüber Beleuchtungsschwankungen. Weiterhin kann der Nutzer nur an den festgelegten Punkten in der Wohnung erkannt werden. Abschnitt 5.3 erkennt den Nutzer an vordefinierten Navigationspunkten ausschließlich mithilfe der Detektionsmodule und setzt ein Verfikationsverfahren ein, um Hypothesen von Nutzern zu bestätigen. Das fortschrittlichste Verfahren aus Abschnitt 5.4 benötigt keine festgelegten Navigationspunkte mehr, sondern durchsucht die gesamte Wohnung explorativ. Ebenfalls wird auf einen mobilen Nutzer und dynamische Hindernisse, wie geschlossene Türen oder verschobene Möbel, reagiert.
In Nutzertests in neun Seniorenwohnungen konnte gezeigt werden, dass das explorative Suchmodul in Kombination mit dem Personentracker, mit den verbleibenden Problemen, eine realwelttaugliche Lösung darstellt. Diese gibt dem Roboter die Möglichkeit, den Nutzer in der Wohnung zu finden, sollte sich dieser nicht im Sichtbereich der Robotersensoren befinden. Das Modul fördert das proaktive Verhalten des Roboters, da dieser aktiv auf den Nutzer zugehen kann, um z. B. Erinnerungen auszuliefern. Eine ähnliche Lösung und umfangreiche Evaluation unter realen Bedingungen ist nach Wissen des Autors auch bis heute nicht in anderen Projekten zur häuslichen Assistenzrobotik zu finden. Verbesserungspotenzial liegt in der Geschwindigkeit der Suche und dem robusten Erkennen von Hypothesen. Ersteres kann beispielsweise durch zusätzliches Priorwissen, wo sich der Nutzer befindet, realisiert werden. Die nicht perfekte Erkennung von Personen, wird aktuell dadurch abgeschwächt, dass der Nutzer mit dem Roboter interagieren kann und beispielsweise durch Dialogeingabe auf dem Touchscreen eine Hypothese des Roboters bestätigen kann. Das Suchmodul profitiert ebenfalls von neuartigen Detektions- und Trackingverfahren.

6. Sturzerkennung

Dieses Kapitel beschreibt ein Verfahren zur Erkennung gestürzter Personen. Die Grundlage bilden Tiefendaten, die mit einem mobilen Roboter gewonnen werden. Die nachfolgenden Erläuterungen basieren größtenteils auf Volkhardt u. a. (2013d) und Schneemann (2013)[1].

6.1. Einleitung

Ungefähr ein Drittel aller Senioren über 65 stürzt mindestens einmal pro Jahr in ihrer Wohnung (Lord u. a. 2003). Da der Sturz und das längere Liegen auf dem Boden ernste gesundheitliche Risiken in sich bergen (Noury u. a. 2008), ist es nicht verwunderlich, dass sich die Sturzerkennung und das Absetzen eines Notrufs unter den meist gewünschten Funktionalitäten der Senioren befinden (Huijnen u. a. 2011).
Aktuell kommerziell verfügbare Lösungen bieten nur eingeschränkte Funktionalität und müssen entweder vom Nutzer am Körper getragen werden oder erfordern bauliche Veränderungen in der Wohnung. Die Mobilität eines Serviceroboters hat den Vorteil, dass der Sturz in nahezu allen Räumlichkeiten nichtinvasiv erkannt werden kann. Da sich der Sturzvorgang außerhalb des Erfassungsbereichs der Robotersensorik ereignen kann, wurde ein Verfahren entwickelt, um gestürzte Personen auf dem Boden zu erkennen. Verdeckungen können teilweise durch das Verfahren beziehungsweise durch die Blickrichtungsänderung des Roboters behandelt werden. Abschließend eröffnet der Mensch-Maschine Dialog ein Nutzerfeedback und weiterführende Informationen für Hilfsstellen, z. B. die Schwere des Sturzes.
Da zur Entwicklungszeit des Verfahrens rein visuelle Verfahren große Probleme mit unterschiedlichen Posen und Verdeckungen aufwiesen, wurde der Algorithmus ausschließlich auf den Tiefeninformationen des Kinect Sensors entwickelt. Im Vergleich zu visuellen Methoden kann die Nutzung von 3D Tiefendaten die Robustheit der Detektion verbessern, indem beispielsweise die Bodenebene geschätzt, eine räumliche Objektsegmentierung durchgeführt wird oder Objekte in ihrer 3D Ausrichtung normalisiert werden.

[1] Vom Autor im Rahmen dieser Arbeit betreut.

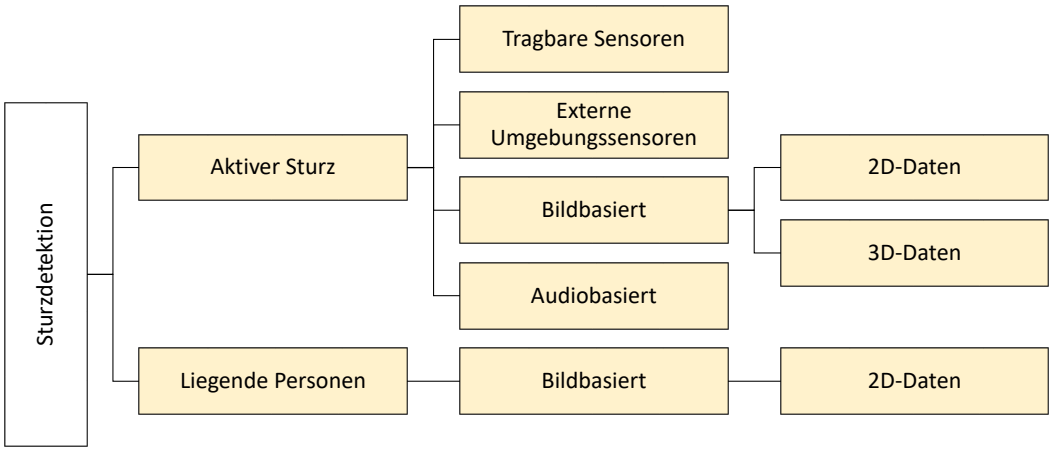


Abbildung 6.1.: Übersicht zu Verfahren der Sturzerkennung. Für die Erkennung liegender Personen existiert kein 3D-Verfahren (Schneemann 2013).

Das entstandene Modul ist prinzipiell unabhängig vom Personentracker, den eingesetzten Detektoren und der Personensuche (Abbildung 2.3). Allerdings kann die Information des Personentrackers (Kapitel 4) genutzt werden, um Falsch-positive auszuschließen oder eine Suche nach gestürzten Personen zu initiieren, wenn lange keine Person in aufrechter Pose erkannt wurde. Weiterhin kann der vorgestellte Suchalgorithmus (Kapitel 5) genutzt werden, um die gesamte Wohnung nach gestürzten Personen abzusuchen.

6.2. Systematisierung der Ansätze

Viele aktuelle Forschungsprojekte befassen sich mit Sturzerkennung, speziell von Senioren. Dennoch besitzen die zurzeit kommerziell verfügbaren Produkte noch Nachteile, die einen breiten Durchbruch verhindert haben.
Abbildung 6.1 gibt einen Überblick über die eingesetzten Methoden zur Sturzerkennung. Etablierte Methoden, die den Sturzvorgang detektieren, nutzen Beschleunigungssensoren, um Beschleunigungen in Richtung Boden oder Abweichungen von gelernten Bewegungsmustern zu erkennen (Philips Electronics 2012). Da diese Sensoren am Körper getragen werden müssen, entsteht die Gefahr des Vergessens durch den Senior, vor allem wenn dieser unter kognitiven Beeinträchtigungen leidet.
Der Sturzvorgang kann weiterhin mittels externer Sensoren, wie Drucksensoren Kameras und Mikrofonen erkannt werden (GmbH Future-Shape 2010; Popescu u. a. 2009). Visuelle 2D und 3D Verfahren können in die Gruppen: Veränderung der Körperform (Vaidehi u. a. 2011; Shoaib u. a. 2011; Planinc

u. a. 2012; Zhang u. a. 2012), der Position (Rougier u. a. 2011) oder der Bewegungsgeschwindigkeit (Mastorakis u. a. 2012) sowie der Analyse von Bewegungsmustern (Anderson u. a. 2006) eingeteilt werden. Für eine ausführliche Übersicht sei auf Willems u. a. (2009) verwiesen. Zur Anwendung der Verfahren sind meist relativ umfangreiche Eingriffe in der Wohnung nötig, um alle Bereiche durch die Sensoren zu erfassen oder es werden Trainingsdaten von echten Sturzsequenzen benötigt.

Andere Verfahren, welche prinzipiell auch auf Robotern zur Anwendung kommen können, detektieren die gefallene Person auf dem Boden. 2D Methoden (Wang u. a. 2011; Lv 2011) benutzen mehrere gedrehte deformierbare partsbasierte Modelle (Abschnitt 3.3.4), um mit verschiedenen Posen, Blickwinkeln und Perspektiven umzugehen. Da diese Methoden jedoch für aufrecht stehende Posen konzipiert wurden, ist die Detektionsrate relativ gering.

Die vorliegende Arbeit bedient sich 3D Personen- und Objektdetektionsverfahren, um ein neuartiges Verfahren zur Detektion von gestürzten Personen auf einem mobilen Roboter zu entwickeln. Sie fokussiert sich auf bekannte featurebasierte Ansätze, während modellbasierte (Xia 2011) oder ausgefallene Verfahren, wie die Nutzung geodätischer Extrema (Plagemann 2010; Schwarz 2011) nicht betrachtet werden.

Featurebasierte Ansätze nutzen Histogrammfeatures (Spinello u. a. 2011b; S. Wu u. a. 2011; Ikemura u. a. 2010a; Hegger u. a. 2012) und geometrische und statistische Features, welche denen aus Abschnitt 3.2.2 ähneln (Spinello u. a. 2010b). Features, die die Oberfläche von Objekten beschreiben (Tang 2011; Fiolka u. a. 2012), speziell die *Histogram of Local Surface Normals* (HLNS) (Hegger u. a. 2012) oder das *Fast Point Feature Histogramm* (FPFH) (Rusu 2009), eignen sich gut, um die unregelmäßigen Formen von gestürzten Personen zu beschreiben.

Zur Behandlung partieller Verdeckungen durch Möbel oder eigene Körperteile eignen sich layer-basierte Methoden (Hegger u. a. 2012; Spinello u. a. 2010b), welche Objekte in mehrere Schichten einteilen. Ein aktuelles Verfahren von Lewandowski u. a. (2017) nutzt 3D Normal Distributions Transform Maps und IRON Features (Schmiedel u. a. 2015), um gestürzter Personen zu erkennen, welche auf Objekte in der Umgebung gefallen sind oder an diese angrenzen.

Zuletzt untersucht diese Arbeit *Interest points* wie SURE und NARF Feature (Fiolka u. a. 2012; Steder u. a. 2010), da diese aufgrund der Invarianz und Vielzahl vielversprechend für die Detektion bei Posenänderung und partieller Verdeckung sind.

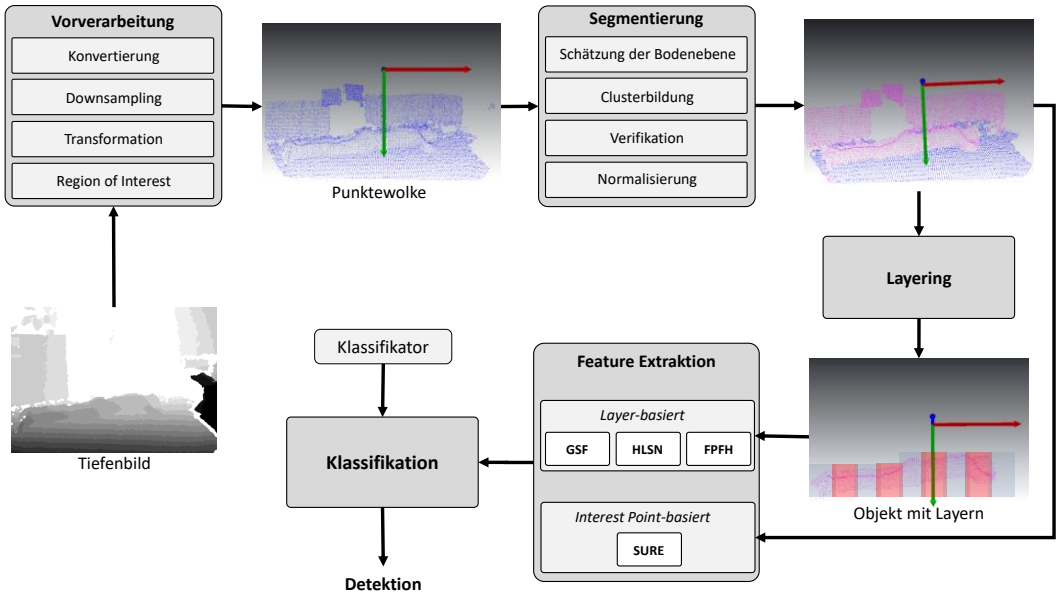


Abbildung 6.2.: Überblick über den vorgestellten Ansatz zur Sturzerkennung mittels Tiefendaten. Der Ansatz besteht aus 5 Phasen (graue Boxen). Erläuterungen: siehe Fließtext (Volkhardt u. a. 2013d).

6.3. Sturzerkennung in Tiefendaten

Das entwickelte Verfahren segmentiert die Punktewolke zunächst in plausible Objektkandidaten, welche anschließend ausgerichtet und in vertikale Schichten (engl. *Layer*) eingeteilt werden. Wird eine Mindestanzahl von Layern von einem Klassifikator positiv klassifiziert, gilt das Objekt als gestürzte Person. Das Verfahren gliedert sich, wie in Abbildung 6.2 sichtbar, in 5 Stufen: Vorverarbeitung, Segmentierung, Layering, Feature-Extraktion und Klassifikation.

6.3.1. Vorverarbeitung

In diesem Schritt wird das Tiefenbild der Kinect in eine vorverarbeitete 3D Punktewolke umgewandelt. Mithilfe der intrinsischen Kameraparameter wird das Tiefenbild in eine 3D Punktewolke konvertiert. Zur Reduktion der Datenmenge wird anschließend ein Voxelgrid eingesetzt, das alle Punkte in den jeweiligen 3D-Boxzellen durch ihren Schwerpunkt repräsentiert. Die Zellgröße beträgt jeweils $3\,cm \times 3\,cm \times 3\,cm$. Nach der intrinsischen Konvertierung transformiert dieser Schritt die Punktewolke anhand der extrinsischen Kameraparameter (Einbauposition der Kinect). Speziell wird der Neigungswinkel der Kinect ausgeglichen, um die spätere Bodenebenenschätzung zu ermögli-

chen. Der Ansatz trifft die Annahme, dass sich gestürzte Personen auf dem Boden befinden. Daher muss nur der untere Teil der Punktewolke betrachtet werden und alle Punkte mit einer Höhe von über $60\,cm$ werden mittels eines PathThrough-Filters entfernt (Region of Interest).

6.3.2. Segmentierung

Die Segmentierungsphase partitioniert die Punktewolke in Objekte, welche als Kandidaten für den Klassifikator dienen. Zunächst werden die Punkte, die zur Bodenebene gehören, mithilfe eines RANSAC-Algorithmus geschätzt (Fischler u. a. 1981). Als Modell wird eine Ebene parallel zur x-y-Ebene des Weltkoordinatensystems angenommen. Punkte, die das aktuelle Modell unterstützen, dürfen einen maximalen Abstand von $10\,cm$ zur Ebene haben und ihre lokalen Oberflächennormalen dürfen nicht mehr als 2° von der Ebenennormalen abweichen. Alle Punkte, die nicht zur Bodenebene gehören, werden anschließend in eine Liste von Clustern $\mathbf{C} = \{c_1, \ldots, c_M\}$ segmentiert. Hierzu kommt ein einfaches Clustering basierend auf dem euklidischen Abstand mit einer Schwelle von $3\,cm$ zum Einsatz.
Auch wenn die Vorverarbeitung bereits alle Punkte außerhalb der ROI herausfiltert, enthält die Punktewolke immer noch Objekte, welche ursprünglich über die ROI hinausragten, da diese Objekte einfach abgeschnitten wurden. Um diese Objekte auszuschließen, werden die Cluster c_i aus $\mathbf{C}$ entfernt, deren Höhe eine geringere Abweichung als $5\,cm$ zur Höhe der ROI besitzen. Alle verbleibenden Cluster in $\mathbf{C}$ gelten als Kandidaten für eine gestürzte Person. Für die folgende Einteilung in Layer wird jedes Objekt in seiner Position und Orientierung ausgerichtet. Hierzu wird der Schwerpunkt eines jeden Punktclusters in den Koordinatenursprung verschoben und das Objekt so gedreht, dass die längste Seite mit der x-Achse übereinstimmt. Der Drehwinkel ergibt sich zwischen der x-Achse und dem Eigenvektor der Kovarianzmatrix des Clusters, dessen Eigenwert maximal ist.

6.3.3. Layering

Eine gestürzte Person kann leicht durch Objekte in der Wohnung oder durch eigene Körperteile partiell verdeckt sein. Daher wird ein schichtenbasierter Ansatz vorgeschlagen. Jedes Cluster c_j wird dazu in eine Sequenz von nebeneinanderliegenden Schichten (Layern) partitioniert: $\mathbf{L}_j = \{l_{j,1}, \ldots, l_{j,K}\}$. Um sicher zu stellen, dass die Layer bei partiellen Verdeckungen und bei unterschiedlichen Personen, die gleichen Körperteile erfassen, wird eine feste Layerbreite anstatt einer festen Layeranzahl, wie es Hegger u. a. (2012) und Spinello u. a. (2010b) für stehende Personen nutzen, verwendet. Käme eine feste Layeranzahl

Tabelle 6.1.: Geometrische und statistische Features.

Feature	Feature
Anzahl der Punkte	Wölbung bzgl. Schwerpunkt
Rundheit	Durchschn. Abweichung vom Median
Flachheit	Normalisierte Residuen bzgl. Flachheit
Linearität	Std. bzgl. Schwerpunkt
Punktedichte	

zum Einsatz, würde die Breite und der Inhalt der Layer stark mit dem Grad der Verdeckung der Person variieren. Um die Layerbreite festzulegen, wurden die Cluster von unverdeckten Personen in acht Layer eingeteilt. Eine Mittelwertbildung ergab eine durchschnittliche Breite von 22.52 cm. Um die Varianz in der Körpergröße unterschiedlicher Personen und die daraus resultierende Varianz der jeweiligen Layerinhalte zu kompensieren, werden überlappende Layer eingesetzt. Es wird eine Überlappung von 2.5 cm zwischen benachbarten Layern verwendet.

6.3.4. Feature-Extraktion

Das Resultat der Segmentierungsphase ist ein Set von 3D Clustern $\mathbf{C}$, wobei jeder Cluster c_j aus einem Set von mehreren Layern $\mathbf{L}_j$ besteht. Während der Feature-Extraktionsphase wird für jeden Layer $l_{j,k}$ ein Featurevektor $f_{j,k}$ berechnet. In der Arbeit wurden vier verschiedene Features untersucht.

Geometrische und statistische Features (GSF)

Der Ansatz nutzt neun geometrische und statistische Features aus Spinello u. a. (2010b); aufgelistet in Tabelle 6.1.

Histogram of Local Surface Normals (HLSN)

Die HLSN bestehen aus einem Histogramm von lokalen Oberflächennormalen und zusätzlichen statistischen 2D- und 3D-Features, um die Charakteristik eines Objekts zu beschreiben. Wie in Hegger u. a. (2012) wird ein separates Histogramm mit sieben Bins für jede Normalenachse (x, y und z) über die Normalen aller Punkte in einem Layer berechnet. Zusätzlich wird die Höhe und Tiefe eines Layers zum Featurevektor hinzugefügt.

Fast Point Feature Histogram (FPFH)

Das FPFH nutzt die Orientierung lokaler Oberflächennormalen, um die Nachbarschaft eines Punktes der Punktewolke zu beschreiben (Rusu 2009). Zur Berechnung des FPFH werden für jeden Punkt die Differenzen zwischen der Orientierung seines Normalenvektors und den Normalenvektoren aller Nachbarpunkte gebildet. Dazu wird ein fixes Koordinatensystem an jedem Punkt definiert und die Differenz zwischen den Normalen durch drei Winkelmaße beschrieben. Im entwickelten Ansatz wird das FPFH für jeden Punkt eines Layers berechnet, anschließend aber der Mittelwert aller FPFHs als Deskriptor des Layers verwendet.

Surface-Entropy (SURE)

Neben dem Layeransatz, untersucht die Arbeit die Nutzung von Interest Points als Features. Diese werden auf dem gesamten Cluster detektiert und anschließend jeweils ein lokaler Deskriptor extrahiert. In Studien erreichten die SURE-Features eine hohe Robustheit bei partiellen Verdeckungen (Fiolka u. a. 2012). Der Keypoint-Detektor und der Deskriptor der SURE- Features beruht auf der Orientierung von Oberflächennormalen. Der Point- Detektor misst die Variation in den Oberflächennormalen und detektiert lokale Maxima. Der Deskriptor ähnelt stark dem FPFH. Da der Interest Point Detektor eine relativ geringe Menge an lokalen Featuredeskriptoren liefert, kann hier auf den Einsatz eines Layerings verzichtet werden.

6.3.5. Klassifikation

Um die extrahierten Features eines Layers beziehungsweise Clusters in die Klassen *gestürzte Person* und *anderes Objekt* einzuteilen, wird ein Klassifikator verwendet. Die Arbeit untersucht vier bekannte maschinelle Klassifikationsalgorithmen: Nearest Neighbor (NN), Random Forest (RF), Support Vector Machine (SVM) und AdaBoost (AB). Die Hyperparameter der Klassifikatoren wurden mittels Cross-Validation variiert (Anzahl der Nachbarn, Anzahl der Bäume, Art des SVM Kernels, etc.) was zu einer Evaluation von 34 Klassifikatoren pro Feature führte (Schneemann 2013). Das Resultat der Klassifikation ist eine Sequenz von Objekten mit jeweils mehreren klassifizierten Layern. Ein Objekt gilt als gestürzte Person, wenn es eine experimentell bestimmte Mindestanzahl von drei positiv klassifizierten Layern enthält.

6.4. Experimentelle Untersuchungen

6.4.1. Datensatz

Zur Gewinnung eines Trainings- und Testdatensatzes wurden Tiefenbilder mit der Kinect des mobilen Roboters erfasst, welche positive und negative Beispiele enthalten. Da eine manuelle Annotation der Objekte sehr aufwendig ist, wurde ein halb automatisches Labeling genutzt. Die negativen Beispiele wurden in fünf unterschiedlichen Wohnungen von Senioren und einem Gemeinschaftsraum, welche zurzeit der Aufnahme frei von Personen waren, gewonnen. Zusätzlich wurden Aufnahmen in einem Testlabor (Anhang A.8) vorgenommen, welche menschenähnliche Objekte, wie Kleidung, Mäntel und Jacken sowie zwei große Hunde auf dem Boden enthielten. Diese Objekte wurden hinzugefügt, da ihre Größe, Form und Oberflächennormalen ähnlich zu liegenden Personen sind und sie somit eine Herausforderung für den Klassifikator darstellen. Für die positiven Beispiele legten sich neun Testpersonen in unterschiedlichen Posen in eine vordefinierte Region auf dem Boden, die keine Objekte oder Verdeckungen enthielt. Da der Fokus der Arbeit auf der Evaluation der besten Feature-Klassifikator Kombination liegt und nicht auf der Segmentierung sind die positiven Trainingsbeispiele relativ einfach vom Hintergrund zu trennen. Durch die fehlenden Verdeckungen befindet sich jedes Körperteil in etwa gleich häufig in den Trainingsdaten und der Klassifikator beschränkt sich nicht auf einen bestimmten Teil des Körpers.
Die Trainingsdaten wurden im Verhältnis von $\frac{2}{3}$ zu $\frac{1}{3}$ in Trainings- und Testdaten eingeteilt. Dabei wurde darauf geachtet, dass die gleichen Objekte (Testpersonen, Räumlichkeiten) nicht sowohl in Trainings- und Testdaten vorhanden waren. Abbildung 6.3 zeigt einige beispielhafte Bilder des Datensatzes.

6.4.2. Evaluationskriterium

Im praktischen Einsatz wiegt eine geringe Falsch-positiv-Rate stärker als eine hohe Richtig-positiv-Rate. Eine hohe Falsch-positiv-Rate würde schnell zum Abschalten des Systems führen. Während der Bewegung durch die Wohnung nimmt der Roboter die Person mehrfach aus unterschiedlichen Blickwinkeln wahr. Der Algorithmus hat somit die Möglichkeit, die gestürzte Person auf einigen Bildern zu verfehlen.
Der $\mathcal{F}$-score bildet das harmonische Mittel aus Precision und Recall und wird zur Bewertung der Klassifikationsgüte genutzt. Der $\mathcal{F}_{0.5}$-score (Rijsbergen 1979) gewichtet die Precision (PR) höher als den Recall (RC), was dazu führt, dass falsch-positiv Detektionen höher (negativ) gewichtet werden, als richtig-positiv

Abbildung 6.3.: Erste und zweite Reihe: positive Trainings- und Testbeispiele. Dritte und vierte Reihe: negative Trainingsdaten aus Seniorenwohnungen und Labor. RGB-Bilder zur Anschaulichkeit. Der Ansatz verwendet ausschließlich Tiefenbilder (Volkhardt u. a. 2013d).

Detektionen:

$$\mathcal{F}_{0.5} = \frac{(1 + 0.5^2) \cdot PR \cdot RC}{0.5^2 \cdot PR + RC}. \tag{6.1}$$

6.4.3. Objektunspezifische Evaluation

Um die allgemeine Güte jeder Feature-Klassifikator Kombination zu bewerten, wurde eine objektunspezifische Evaluation durchgeführt: Die Featurevektoren aller Layer beziehungsweise Interest Points in den Testdaten wurden klassifiziert und der $\mathcal{F}_{0.5}$-score berechnet, ohne die Objektzugehörigkeit der Featurevektoren zu berücksichtigen.
Abbildung 6.4 zeigt die Ergebnisse der objektunspezifischen Evaluation der jeweiligen Feature-Klassifikator Kombinationen. Der AdaBoost Algorithmus erreicht mit allen vier Features die schlechteste Klassifikationsgüte. Der Grund liegt in der Beschränkung auf eindimensionale Weak-Learner, welche auch in Arras u. a. (2007) verwendet werden. Zukünftige Arbeiten könnten Entscheidungsbäume (siehe Abschnitt 3.2.2) einsetzen, um die Güte zu erhöhen. Die Betrachtung der Features zeigt, dass die HLNS allen anderen Features überlegen sind. Weiterhin ist der geringe $\mathcal{F}$-score der SURE Features auffällig. Da der Deskriptor der SURE Feature ähnlich zum FPFH ist, welches gute Ergebnisse zeigt, ist die Ursache in der Instabilität der detektierten Interest Keypoints zu finden. Wie in Abbildung 6.5 sichtbar, schwankt die Anzahl und die Position der detektierten Keypoints zwischen unterschiedlichen Personen. Ebenfalls erkennbar ist, dass das gleiche Problem auch bei derselben Person in leicht unterschiedlicher Pose auftritt.

6.4.4. Objektspezifische Evaluation

Im vorherigen Abschnitt wurden die besten Feature-Klassifikator Kombinationen ermittelt. In diesem Abschnitt werden diese Kombinationen genutzt und analysiert, wie viele Layer beziehungsweise Interest Points pro Objekt positiv klassifiziert werden müssen, um ein Objekt als gestürzte Person zu erkennen und den besten $\mathcal{F}_{0.5}$-score über alle Objekte zu erzielen.
Die Ergebnisse bestätigen die Resultate der objektunspezifischen Evaluation. Die HLSN übertreffen erneut alle anderen Features. Abbildung 6.6(a) zeigt die Precision-Recall-Kurve. Die durchschnittliche Präzision (engl. *average precision*, AP), berechnet durch die Integration der Fläche unter jeder Kurve, bestätigt die Bewertungen auf Basis des $\mathcal{F}_{0.5}$-scores. Die unterschiedlichen Punkte der Kurve wurden generiert, indem die Anzahl der Layer beziehungsweise Interest Points, welche positiv klassifiziert werden müssen, variiert wurden, damit ein Objekt als gestürzte Person gilt.

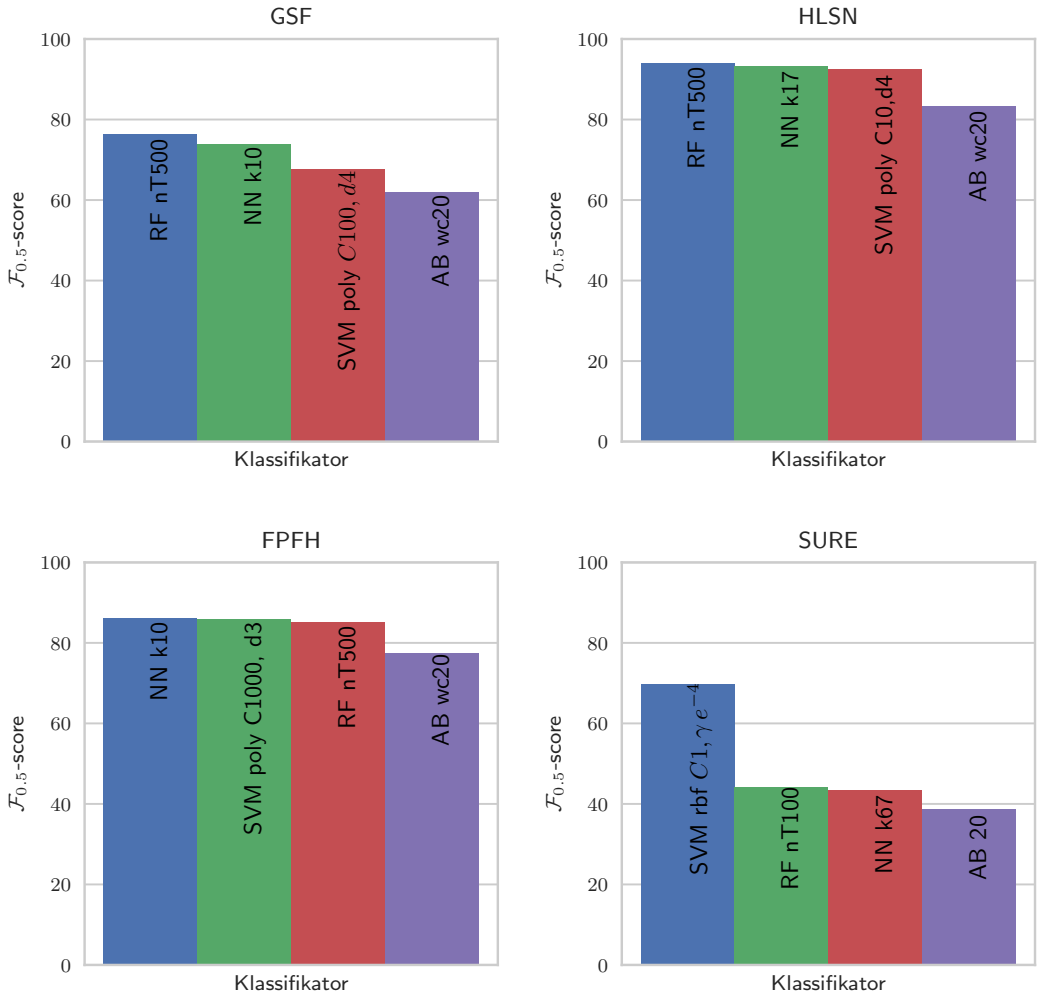


Abbildung 6.4.: Objektunspezifische Evaluation: $\mathcal{F}_{0.5}$-score der untersuchten Features mit verschiedenen Klassifikatoren und den besten cross-validierten Hyperparametern. k-Nearest Neighbor (NN), Random Forest (RF), Support Vector Machine (SVM) und AdaBoost (AB) (Volkhardt u. a. 2013d).

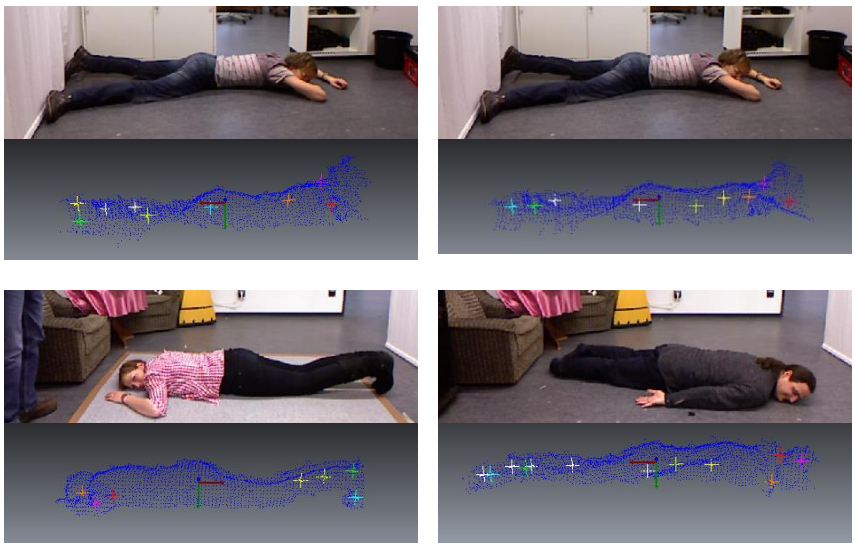
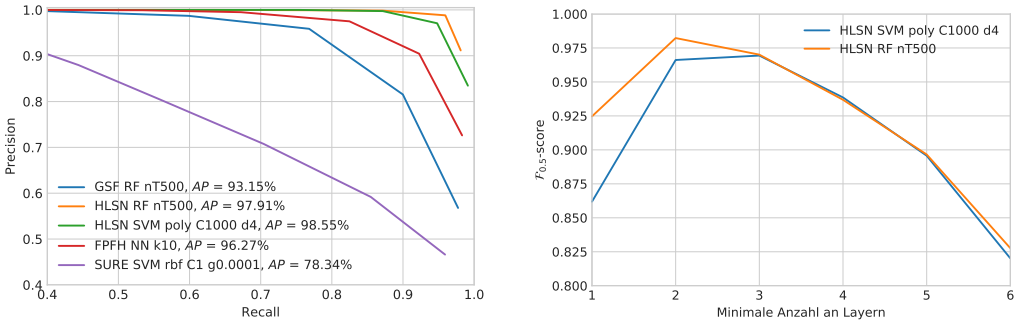


Abbildung 6.5.: Interest Point: Die Anzahl und die Position der detektierten Interest Points schwankt auch auf ähnlichen Ansichten stark (Volkhardt u. a. 2013d).



(a) Objektspezifische Evaluation **(b)** HLSN-Klassifikatoren

Abbildung 6.6.: (a) Precision-Recall-Kurven unterschiedlicher Features und Klassifikatoren mit zugehöriger average precision (AP). (b) HLSN: Vergleich zwischen dem Klassifikator mit der höchsten Klassifikationsgüte (RF) und dem performantesten Klassifikator (SVM).

6.4.5. Laufzeit

Für den praktischen Einsatz des Verfahrens wird eine echtzeitfähige Implementierung auf dem Roboter benötigt. Daher wurden die Laufzeiten der Featureberechnung und der Klassifikatoren untersucht und verglichen. Die beste Erkennungsleistung wird mit der Kombination aus HLSN Features und einem Random Forest Klassifikator mit 500 Bäumen erreicht. Die benötigte Zeit zur Klassifikation eines Objektes beträgt damit $20.16\,ms$, was bei einer Vielzahl an Objekten in der Szene den echtzeitfähigen Einsatz verhindert. Im Gegensatz dazu benötigt eine Support Vektor Maschine mit einem polynomiellen Kernel mit Rang $d = 4$ und einem Strafterm $C = 1000$ nur $0.178\,ms$ pro Objektklassifikation. Da die Differenz der Klassifikationsgüte, wie in Abbildung 6.6(b) sichtbar, zwischen RF und der SVM vertretbar gering ist, werden die HLSN im finalen System mit einer SVM kombiniert. Dies erlaubt einen echtzeitfähigen Einsatz auf dem mobilen Roboter.

6.4.6. Klassifikationsgüte des finalen Systems

Wie in Abbildung 6.6(b) sichtbar, werden die besten Ergebnisse mit der HLSN-SVM Kombination erzielt, wenn die Schwelle für eine positive Detektion bei drei positiv klassifizierten Layern liegt.

Die vorgeschlagene Feature-Klassifikator Kombination führt auf dem Testdatensatz zu einem $\mathcal{F}_{0.5}$-score von 96.94 % und einer Accuracy von 96.08 % mit Recall 87.17 % und Precision 99.74 %. Die Falsch-positiv-Rate liegt bei 0.1 %. Dies entspricht in einem angenommenen Datenstrom von 10 Bildern pro Sekunde mit jeweils durchschnittlich fünf zu klassifizierenden Objekten pro Frame einer Falsch-positiv Klassifikation pro 20 Sekunden. In den Datensätzen mit den Tiefendaten der Seniorenwohnungen traten keine falsch-positiv Detektionen auf, was auf die strikte Vorverarbeitung und die gewählte Feature-Klassifikator Kombination zurückzuführen ist. Da der Recall nicht perfekt ist, wird eine gestürzte Person nicht in jedem Frame erkannt. Die finale Detektionsleistung wird jedoch durch die Tatsache erhöht, dass der Roboter durch die Wohnung fährt und die Person aus unterschiedlichen Blickwinkeln betrachten kann. Genauere Analysen über die Klassifikationsgüte im tatsächlichen Einsatz, z. B. Richtig- und Falsch-positiv-Rate bei längerfristig beobachteten Objekten oder die Klassifikationsgüte bei zeitlicher Filterung mehrerer Detektionen, sowie die Kombination mit anderen Modulen sind jedoch Bestandteil zukünftiger Arbeiten.

6.5. Diskussion und Fazit

Dieses Kapitel präsentierte eine Methode, um gestürzte Personen zu erkennen. Das featurebasierte Verfahren (Volkhardt u. a. 2013d) war zur Veröffentlichung auch international das Erste, das gestürzte Personen ausschließlich rein tiefenbasiert durch die Verwendung von 3D Daten der Kinect erkennt und auf einem mobilen Roboter lauffähig ist.
Zunächst wird die 3D-Punktewolke in Objekte segmentiert und in Schichten eingeteilt, um mit partieller Verdeckung umzugehen. Anschließend werden Features extrahiert und ein trainierter Klassifikator verwendet, um die Schichten beziehungsweise Objekte in Personen und Nicht-Personen einzuteilen. In experimentellen Untersuchungen wurde gezeigt, dass die Kombination aus *Histograms of Local Surface Normals* und einer SVM die besten Ergebnisse hinsichtlich Laufzeit und Klassifikationsgüte liefern. In weiterführenden Arbeiten kann die auf dem euklidischen Abstand basierende Segmentierung verbessert werden. Stürzen Personen auf oder in der Nähe von Möbeln, enthalten die segmentierten Objekte manchmal Teile des Möbelstücks. Dies lässt sich beispielsweise durch die Integration von RGB Informationen oder anderen Repräsentationsformen der Tiefendaten (Lewandowski u. a. 2017) verbessern. Weitere Verbesserungsmöglichkeiten umfassen die Verwendung von mehreren spezialisierten Klassifikatoren pro Schicht, die ähnlich dem ISM Ansatz (Leibe u. a. 2008) für die Objektmitte abstimmen. Weiterhin muss das Verfahren in Kombination mit den anderen Modulen im täglichen Einsatz in den Seniorenwohnungen untersucht werden. Erste Ansätze zur Verbesserung des Verfahrens sind in (Lewandowski 2016; Wengefeld u. a. 2016; Lewandowski u. a. 2017) zu finden.

7. Einsatz im häuslichen Szenario

In diesem Kapitel wird der Einsatz der entwickelten Architektur zur Personenwahrnehmung im realen Anwendungsszenario beschrieben. Im Rahmen des SERROGA Projekts (SERROGA 2012) wurde eine Fallstudie durchgeführt, welche die Akzeptanz des Serviceroboters und die persönlichen Präferenzen der Nutzer untersuchte (Doering u. a. 2015). Anschließend verblieb der Roboter zu Evaluationszwecken bis zu 3 Tage in den Wohnungen von 3 Projektmitarbeitern beziehungsweise 9 Senioren (Gross u. a. 2015). In Abbildung 7.1 ist eine beispielhafte Aufnahme aus den Nutzertests dargestellt. Die gesamte Testdauer umfasste 16 Tage beziehungsweise 120 Nettostunden. Zuvor verblieb der Roboter für Funktionstests 7 Tage in den Wohnungen von Projektmitarbeitern. Im Rahmen der Nutzertests unter Realweltbedingungen stand der Roboter und alle seine Funktionalitäten zur freien Nutzung zur Verfügung. Die Nutzer nahmen im Verlauf der Versuche, je nach persönlicher Präferenz, verschiedene Dienste des Roboters in Anspruch, z. B. Videotelefonie mit automatischer Suche bei eingehenden Anrufen, Terminerinnerungen, Rufen des Roboters zur aktuellen Position (Suche) und die Funktion „Folge mir“. Abbildung 7.2 zeigt einen Ausschnitt der, durch einen Senior individuell genutzten, Roboterfunktionalitäten während der Nutzertests.
Der **Personentracker** mit den zugehörigen **Detektoren** erreichte ein technisch robustes Niveau, sodass sich während der Tests auf die eigentlichen Funktionalitäten des Roboters, wie Videotelefonie, Folgen und Vitalparametermessung konzentriert werden konnte. Auch wenn, bis auf die indirekte Evaluation des Personentrackers im Folgeverhalten (Abschnitt 4.10.5), keine quantitative Untersuchung in den Seniorenwohnungen durchgeführt wurde, zeigt die hohe Akzeptanz der Senioren (Gross u. a. 2015) die Robustheit der technischen Roboterfunktionalitäten. Vor allem die Fusion unterschiedlicher Detektoren im Personentracker tragen dazu bei, dass der Nutzer in verschiedenen Posen unter den anspruchsvollen und wechselnden Beleuchtungsbedingungen einer Seniorenwohnung erkannt und getrackt werden kann (Abbildung 2.2(b) und Anhang A.8). Die quantitative Evaluation des Personentrackers wurde in einer nachgestellten Seniorenwohnung durchgeführt Abschnitt 4.10. Die Personenerkennung kann durch neuere Detektionsverfahren (Abschnitt 3.3.6), neue Sensorik und Trackingverfahren verbessert werden. Vor allem personenähnliche Objekte, ungewöhnliche Posen oder schlechte Lichtverhältnisse führen mit

Abbildung 7.1.: Roboter während der Nutzertests in einer Seniorenwohnung (Gross u. a. 2015).

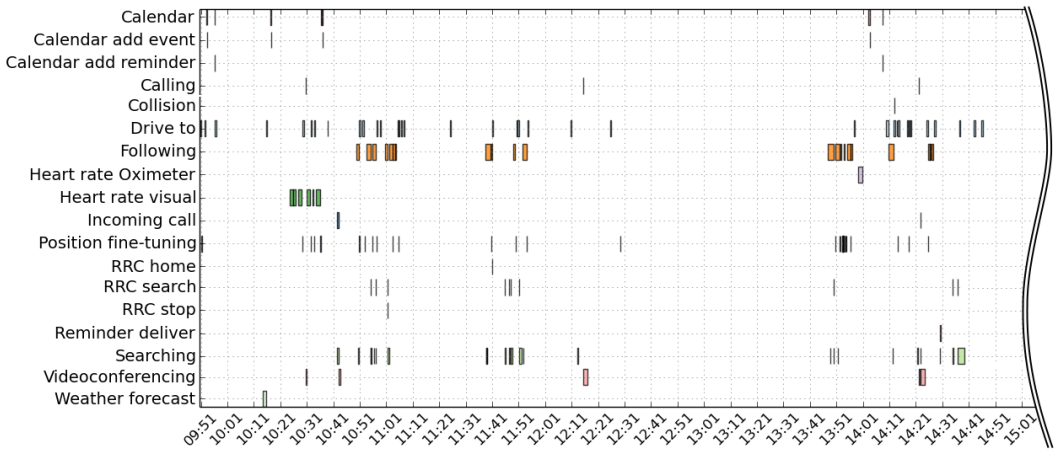


Abbildung 7.2.: Ausschnitt des Nutzungsverhaltens eines Seniors. Die Balken geben den Zeitpunkt und die Dauer der genutzten Services des Roboters an.

dem aktuellen System noch immer zu falsch-positiv und falsch-negativ Detektionen.
Die **Personensuche** kam bei eingehenden Videoanrufen, Terminerinnerungen und Fernbedienungsrufen zur Anwendung (Abbildung 7.2). Die Experimente aus den Nutzertests (siehe Abschnitt 5.4.3) verdeutlichen, dass das System realwelttauglich ist. Die Personensuche profitiert direkt von besseren Detektoren und Trackingalgorithmen. Die Suchstrategie selbst könnte beispielsweise durch ein lernendes System, das sich individuell an die Gewohnheiten des Nutzers und die Gegebenheiten der Wohnung anpasst, verbessert werden. Weiterhin sichert eine robustere Navigation, Hinderniswahrnehmung und Zielanfahrt, dass sich der Roboter schnell und ohne Kollision zum Nutzer bewegen kann (Gross u. a. 2015).
Die **Sturzerkennung** wurde nicht direkt mit Senioren evaluiert. Die Experimente in Abschnitt 6.4 zeigen jedoch, dass der Ansatz vielversprechende Ergebnisse unter Laborbedingungen und in den (leeren) Seniorenwohnungen liefert. Für eine Evaluation mit echten Nutzern sollten die Verbesserungen aus Abschnitt 6.5 umgesetzt und das Modul in den Dialogablauf des Roboters integriert werden.
Durch die modulare **Architektur** konnte das System im Verlauf des Projekts (2012-2015) einfach und zeitnah durch neu entwickelte Detektionsmodule (z. B. Abschnitte 3.2.2 und 3.3.4) ausgerüstet werden. Weiterhin erlaubt die Architektur eine sinnvolle Kombination der Funktionalitäten der einzelnen Komponenten. Verliert der Personentracker den Nutzer beispielsweise während des Folgeverhaltens für mehrere Sekunden, wird eine lokale Suche gestartet, welche es dem Roboter erlaubt, den Nutzer wieder in den Sensorsichtbereich zu bringen und zu erfassen. Sobald der Nutzer erneut erkannt wird, wechselt der Roboter automatisch wieder in den Folgemodus. In Zukunft können weitere Module miteinander interagieren. Beispielsweise kann die Sturzerkennung die Suchstrategie der Personensuche nutzen oder Falsch-positive ausschließen, wenn der Personentracker eine stehende Person erkennt.
Zuletzt zeigt sich die Robustheit des Systems dadurch, dass in, an die Nutzertests anschließenden, Interviews die Leistung der entwickelten Grundmodule (Navigation, Tracking, Suche) von keinem Senior bemängelt wurde. In zukünftigen Projekten und Nutzertests sollte der Roboter vor allem durch neue (Gesundheits-) Services verbessert werden und das System in längeren Nutzertests evaluiert werden (Gross u. a. 2015). Die in dieser Arbeit entwickelten Module umfassen bereits einen nennenswerten Anteil am Gesamtsystem und erlaubten es, dem Roboter in realen Wohnungen proaktiv mit den Nutzern zu interagieren.

8. Zusammenfassung und Ausblick

Es folgt eine Zusammenfassung der Kapitel dieser Arbeit, wobei deren wesentliche Beiträge in konzentrierter Form formuliert werden. Zuletzt werden Ideen zur Fortsetzung dieser Arbeit beschrieben.

8.1. Zusammenfassung

Die Arbeit stellte eine Architektur zur Nutzerwahrnehmung für mobile Roboter in häuslichen Szenarien vor (Kapitel 2). Die Systemarchitektur definiert die beteiligten Komponenten zur Nutzerwahrnehmung und deren Kommunikation untereinander. Speziell werden asynchrone Detektionsmodule, ein Personentracker, eine Komponente zur Personensuche und ein Modul zur Sturzdetektion eingesetzt. Ein Vergleich mit anderen Projekten der Assistenzrobotik ergab, dass bisher keine andere Architektur alle für das Anwendungsszenario nötigen und von den Senioren gewünschten Kriterien an einen mobilen Roboter erfüllt. Die entwickelte Architektur füllt diese Lücken und stellt ein realwelttaugliches Gesamtkonzept vor, das sich in mehrtägigen Tests bewährt hat. Durch die modulare Entwicklung und die Definition einer gemeinsamen Schnittstelle arbeiten die beteiligten Module unabhängig und können einfach ausgetauscht und erweitert werden (Abschnitt 2.3).
Die Arbeit untersuchte eine Vielzahl an Detektionsalgorithmen in Laser-, Bild- und 3D-Tiefendaten (Kapitel 3). Für das Anwendungsszenario geeignete Algorithmen wurden ausgewählt, angepasst und verbessert. Eine spezielle Herausforderung des Anwendungsszenarios ist die Erkennung des Nutzers in verschiedenen Posen und unter variablen Beleuchtungsbedingungen. Dies wurde gelöst, indem einerseits speziell auf die Erkennung verschiedener Körperposen spezialisierte Detektoren eingesetzt wurden und andererseits mehrere Detektoren kombiniert wurden. Aufgrund der gemeinsamen Schnittstelle zum Personentracker liegt das Sensormodell in den jeweiligen Detektoren (Abschnitt 3.5). Die generierten Hypothesen werden unter Berücksichtigung von unsicheren Transformationen in Weltkoordinaten transformiert.
Die asynchronen Hypothesen der Detektoren werden von einem Personentracker raumzeitlich gefiltert, um robuste Personentracks zu erhalten (Kapitel 4). Der Personentracker ermöglicht den Einsatz verschiedener Filteralgorithmen

und Systemmodelle, welche dynamisch ausgetauscht werden können. Dies ermöglicht eine schnelle Entwicklung und Evaluation unterschiedlicher Modelle im betrachteten und weiteren Szenarien (Weinrich u. a. 2012; Weinrich u. a. 2013b). Die Datenassoziation nutzt Covariance Intersection für abhängige Beobachtungen, und verspätete Beobachtungen werden mittels Rückwärtsprädiktion integriert (Abschnitt 4.7). Für jede Hypothese schätzt der Personentracker eine Existenzwahrscheinlichkeit. Ferner wurden ein Hypothesenmanagement und die Nutzung von optionalem Wissen in Form von Umgebungskarten implementiert. Die Evaluation untersuchte die Detektoren sowie den Personentracker mit verschiedene Detektorkombinationen auf realistischen häuslichen Datensätzen (Abschnitt 4.10). Die besten Ergebnisse konnten mit einem Beinpaardetektor in Kombination mit einem deformierbaren parts-basierten Detektor (DPM) erreicht werden. Für den Einsatz im Anwendungsszenario wurde ein linearer Kalman-Filter mit einem auf konstanter Geschwindigkeit basierenden Systemmodell ausgewählt (Volkhardt u. a. 2013a). Im Anwendungsszenario wurde der Personentracker implizit über das Folgeverhalten untersucht. Schwächen des Systems umfassen Falsch-negative bei schwierigen Posen und Falsch-positive bei personenähnlichen Objekten.
Die vorgestellten Verfahren zur Personensuche erlauben es dem Roboter, den Nutzer in der Wohnung zu suchen, sobald dieser den Sensorsichtbereich verlassen hat (Kapitel 5). Durch die explorative Suchstrategie sucht der Roboter die Wohnung effektiv ab (Abschnitt 5.4). Sobald der Nutzer gefunden wird, bringt sich der Roboter in die optimale Interaktionsposition. Das Verfahren kann mit Falsch-positiven, dynamischen Hindernissen und einem mobilen Nutzer umgehen. In Nutzertests in häuslichen Wohnungen konnte in 98 Suchfahrten eine Erfolgsrate von 86 % erzielt werden. Eine ähnliche Lösung und umfangreiche Evaluation unter realen Bedingungen existiert nach Wissen des Autors noch nicht bei einem anderen Projekt.
Die Sturzerkennung erlaubt es dem Roboter, am Boden liegende Personen auf Basis der 3D-Tiefendaten der Kinect Kamera zu erfassen (Kapitel 6). Das Verfahren segmentiert die 3D-Punktewolke in Objekte und teilt diese anschließend in Schichten ein, um mit partieller Verdeckung umzugehen. Anschließend werden Features extrahiert und ein trainierter Klassifikator verwendet, um Personen zu erkennen. In Untersuchungen mit verschiedenen Features und Klassifikatoren wurde die Kombination aus Histograms of Local Surface Normals und einer SVM ausgewählt, um die besten Ergebnisse hinsichtlich Laufzeit und Klassifikationsgüte zu erreichen. Die vorgeschlagene Kombination führt auf dem Testdatensatz zu einem $\mathcal{F}_{0.5}$-score 96.94 %. Das entwickelte Verfahren war das erste, das gestürzte Personen ausschließlich rein tiefenbasiert mithilfe einer Kinect auf einem mobilen Roboter erkennt.
Im Anwendungsszenario verblieb der Roboter jeweils bis zu 3 Tage in den Woh-

nungen von Senioren (Kapitel 7). Die gesamte Testdauer umfasste 16 Tage. Im Rahmen dieser Nutzertests unter Realweltbedingungen stand der Roboter und alle seine Funktionalitäten zur freien Nutzung zur Verfügung. In an die Nutzertests anschließenden Umfragen wurde keines der vorgestellten Module von den Senioren bemängelt. Zusammenfassend konnte die Arbeit einen umfangreichen Beitrag zum gesetzten Ziel, der Entwicklung von effizienten Methoden zur Nutzerwahrnehmung in häuslichen Umgebungen, beitragen und somit die Markteintrittsbarrieren von Assistenzrobotern senken. Dennoch besteht Verbesserungs- und Erweiterungspotenzial.

8.2. Ausblick

Zukünftigen Arbeiten können an mehreren Stellen der Arbeit anknüpfen. Neuartige Detektionsverfahren können die Erkennungsleistung weiter verbessern. Vor allem Deep Learning Verfahren haben ein hohes Potenzial Personen in variablen Posen robust zu erkennen (Abschnitt 3.3.6). Dichte Punktewolken und Tiefenbilder liefern zusätzliche Informationen und sind invariant gegenüber Beleuchtungsschwankungen (Abschnitt 3.4.1). Ebenfalls stellt die Anwendung von Deep Learning Verfahren auf Tiefendaten einen wenig erforschten Bereich der Personenerkennung dar.
Die gemeinsame Schnittstelle der Architektur kann um weitere Informationen erweitert werden, falls die Detektoren weitere Eigenschaften von Personen, wie beispielsweise Blickrichtung, Pose, Aktivität oder Emotionen, liefern.
Der Personentracker kann, je nach Anwendungsszenario, andere Systemmodelle verwenden, um weitere Informationen zu verarbeiten. Die Existenzschätzung kann durch Detektionsmodelle verbessert werden. Diese definieren, wo und wie sicher ein Detektor Personen im Sensorsichtbereich erkennen kann (Schubert 2011; Richter 2012). Weiterhin können Nutzereigenschaften, wie typische Kleidung und Präferenzen, persistent getrackt und gespeichert werden. Erste Ansätze liefern Volkhardt (2008) und Eisenbach u. a. (2015).
Die Personensuche kann durch Vorwissen über die historischen Aufenthaltsorte des Nutzers beschleunigt werden. Weiterhin profitiert die Suche von einem verbesserten Nutzertracking.
Auch die Sturzdetektion bietet Potenzial für Verbesserungen. Erste Ansätze hierfür liefern Wengefeld u. a. (2016) und Lewandowski u. a. (2017). Die Verfahren nutzen neben einer verbesserten Segmentierung, wenn die Person nah an anderen Objekte liegt, neuartige Features und Detektionsverfahren, welche die Erkennungsleistung unter stärkeren Verdeckungen erhöht. In zukünftigen Arbeiten ist es sinnvoll, die RGB-Informationen einer Kamera mit den Tiefendaten zu kombinieren. Auch hier können Deep Learning Verfahren

untersucht werden. Eine engere Verzahnung mit dem Personentracker und der Personensuche ermöglicht den Ausschluss von Falsch-positiven und sinnvolle Suchstrategien.
Zuletzt muss der Roboter mit einer noch größeren Nutzerzahl und über einen längeren Zeitraum evaluiert werden. Hierfür bedarf es aber neben den oben beschriebenen Verbesserungen auch an zusätzlichen Services mit Mehrwert für den Nutzer, wie beispielsweise kognitive und körperliche Fitnessübungen und bessere Kommunikationsmöglichkeiten (Gross u. a. 2015).
Die in dieser Arbeit entwickelten Methoden und die vorgeschlagenen Verbesserungen unterstützen das Ziel, dass in Zukunft immer mehr Senioren ihren Lebensabend unabhängig zu Hause verbringen können.

A. Anhang

A.1. Roboterplattform

Die verwendete Roboterplattform „Max“ wurde im Projekt CompanionAble[1] als Prototyp entwickelt und im Projekt SERROGA[2] nutzer- und bedarfszentriert verbessert. Der Fokus des Assistenzroboters liegt dabei auf der sozialen statt physischen Interaktion mit dem Nutzer (Tapus u. a. 2007). Abbildung A.1 zeigt den Roboterprototyp und dessen Sensoren und Aktuatoren. Beim Design wurde auf eine relativ geringe Größe (1,2 m) mit kleiner Grundfläche (0,5 m ∅) sowie ein geringes Gewicht (40 kg) geachtet, um eine agile Navigation in der Wohnung zu ermöglichen. Die wichtigsten Sensoren umfassen eine hochauflösende 2 Mp Kamera mit einem 180° Fischaugenobjektiv zwischen den Augen, eine Kinect 3D Tiefenkamera oberhalb des Displays, einen SICK S300 Laserscanner mit 270° Öffnungswinkel, ein Mikrofon auf der Oberseite des Kopfes, einen Kollisionssensor als Notstopp (Bumper) und einen neigbaren Touchscreen zur Nutzereingabe. Die Aktuatoren bestehen aus dem Touchscreen zur Darstellung von Inhalten, Lautsprechern im Kopf und zwei OLED-Displays als Augen, um Emotionen auszudrücken. Weiterhin besitzt der Roboter eine Ablagebox mit integriertem RFID Lesegerät, um persönliche Dinge des Nutzers zu verwahren (Gross u. a. 2012). Der Roboter wird durch einen on-board PC mit Intel i7-620M quad core Prozessor und 8 GB RAM gesteuert. Im Laufe des SERROGA Projekt wurde ein zweiter Intel i7 Prozessor hinzugefügt. Die Energieversorgung wird durch ein automatisches Ladesystem gesichert, dass es dem Roboter erlaubt, seine Akkus bei niedrigem Energiestand oder über Nacht zu laden. Bei der Entwicklung wurde auf einen geringen Preis geachtet, sodass ein Manipulator in Form eines Armes sowie kostspielige Sensoren, wie Laser-Arrays, Time-of-Flight-[3] und Wärmebildkameras fehlen.
Die Dienste des Roboters umfassen Terminverwaltung, Vitalparametermessung, kognitive und physische Übungen, Videotelefonie, Fernsteuerung durch

[1] http://www.companionable.net/
[2] http://www.serroga.de/
[3] Time-of-flight Kameras waren während der Projektdauer nur in nicht vertretbaren Preissegmenten verfügbar. Mit der Einführung der Kinect 2 und ähnlichen Modellen kann ein Einsatz erneut evaluiert werden.

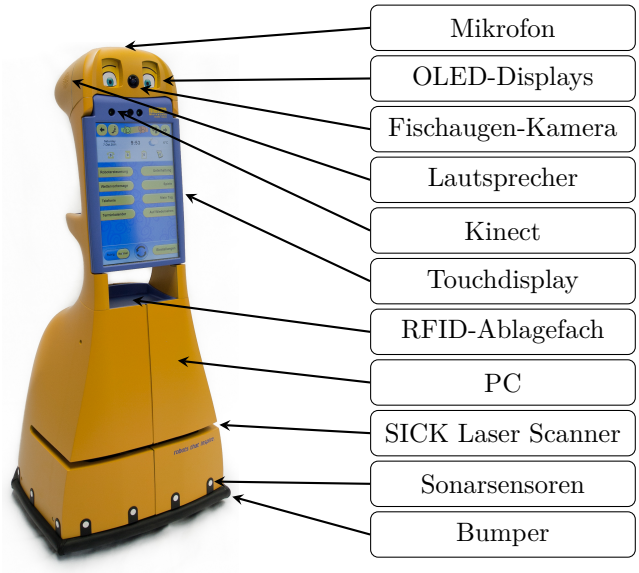


Abbildung A.1.: Roboterplattform „Max“. Die Abbildung zeigt den Roboter und dessen Sensoren und Aktuatoren.

autorisierte Personen, Nutzersuche, Sturzerkennung sowie Entertainmentfunktionen, wie Radio, Bildergalerien oder Hörbücher. Diese Dienste stellen eine erste Auswahl dar und sollen in Zukunft Stück für Stück erweitert werden.

A.2. Projekte der Assistenzrobotik

Dieser Abschnitt listet ausgewählte Projekte der Assistenzrobotik und vergleicht diese mit dem Rahmenprojekt der Dissertation nach Kriterien, die den Kernpunkten der Arbeit entsprechen. Anhang A.2 beschreibt den Inhalt der Projekte und deren technische Funktionalitäten. Ein „x“ kennzeichnet, dass das Kriterium voll erfüllt wird, ein „-“ deutet an, dass das Kriterium nicht erfüllt wird.

	Autonomie	**Pers. Erkennung**	**Folgen**	**Suche**	**Sturzerkennung**	**Einsatzumgebung**
SERROGA (2012-2015) (2012)						
Einsatz	Der entwickelte Roboter „Max“ unterstützt Senioren in ihren Wohnungen mittels Telekommunikation und Telepräsenz, kognitiven und physischen Übungen sowie Terminverwaltung, Vitalparametermessung, Sturzerkennung und Entertainmentfunktionen.					

wird auf der nächsten Seite fortgesetzt

Fortsetzung der letzten Seite

	Autonomie	**Pers. Erkennung**	**Folgen**	**Suche**	**Sturzerkennung**	**Einsatzumgebung**
Kriterien	Autonom	x	x	x	x	Labor / Wohnung
techn. Funktionalität	Der Roboter besitzt ein autonomes Verhalten (proaktive Erinnerungen, Lademechanismus) und kann frei in der Umgebung navigieren. Das System nutzt in dieser Arbeit entwickelte Methoden zur Nutzerwahrnehmung, Nutzersuche, Sturzdetektion und dem Folgeverhalten. Alle Algorithmen laufen ausschließlich auf dem Roboter. Das System wurde in Laborumgebungen und realen Seniorenwohnungen evaluiert.					

ExCite - Telepresence with Giraff (2010-2014) (ExCITE 2010)

Einsatz	Seit 2010 wird Giraff in 15 Haushalten von älteren Personen getestet. Diese sind in der Regel alleinstehend und können so Kontakt zu Arzt und Verwandten halten. Neben dem sozialen Aspekt, dass die Nutzer ihre Verwandtschaft sehen und mit ihnen sprechen können, kann der Arzt regelmäßig Blutdruck, Zucker-Wert und andere kleine Untersuchungen durchführen lassen, deren Ergebnisse direkt per Mail versandt werden. Die Verbesserung GiraffPlus kombiniert den Roboter mit einem, in der Wohnung und am Roboter installierten, Sensornetzwerk, um Vitalparameter und Hauszustände zu erfassen (GiraffPlus 2012).					
Kriterien	Ferngesteuert	-	-	-	-	Wohnung
techn. Funktionalität	Während einer Videokonferenz kann der Roboter frei im Raum bewegt werden. Die Steuerung erfolgt jedoch komplett manuell. Zum einen kann der Besitzer den Roboter durch eine Fernbedienung steuern, zum anderen können Angehörige mithilfe einer Software den Roboter aus der Ferne steuern. Dabei sehen sie auf dem Monitor die Umgebung. Mithilfe der Maus kann der Roboter in die gewünschte Richtung gelenkt werden oder die Displayhöhe variiert werden.					

VGo (VGo 2011)

Einsatz	VGo ist ein Roboter, welcher dazu dienen soll, dass Personen Kontakt mit anderen Mitmenschen haben, auch wenn diese nicht direkt in der Nähe sind. Die Anwendungsbereiche umfassen Produktion, Gesundheitspflege und Bildung.					
Kriterien	Teilautonom	-	-	-	-	Gemeinschaft
techn. Funktionalität	Die Steuerung des Roboters erfolgt durch externe Personen, welche ebenfalls einen VGo besitzen müssen. Dabei kann ein Nutzer bei einem Anruf den Roboter steuern. Hierbei sind Kamera, Mikrofon und Display integriert. Die Steuerung erfolgt manuell oder durch Klicken auf einen Standort. Des Weiteren wird der Nutzer visuell über die Verfügbarkeit der anderen Nutzer informiert. Wenn die Batterie leer ist, meldet dies der Roboter und durch einen Tastendruck kann er zu seiner Dockingstation gesendet werden.					

wird auf der nächsten Seite fortgesetzt

Fortsetzung der letzten Seite

	Autonomie	Pers. Erkennung	Folgen	Suche	Sturzerkennung	Einsatzumgebung

MobiServ (Mobiserv 2009)

Einsatz	Der soziale Assistenzroboter Kompai unterstützt ältere Personen in ihren Wohnungen und kombiniert Smart-Home Sensorik für Vitalparametermessung, kognitive und physische Übungen, Telekommunikation, Tagesablaufstrukturierung, Schlafanalyse und Sturzerkennung. Weiterhin sollen Aktivitäten, Trinkgewohnheiten und Gefahrensituationen analysiert werden.					
Kriterien	Autonom	x	-	-	(x)	Wohnung
techn. Funktionalität	Der Roboter kann Personen mittels Kamera und Kinect wahrnehmen und nimmt Kommandos über Sprache und Touchscreen entgegen. Die Sturzerkennung erfolgt mittels externer getragener Sensorik. Der Roboter ist in der Lage selbstständig an eine Ladestation anzudocken.					

PR2 (PR2 2010)

Einsatz	Bei Willow Garages PR2 handelt es sich um einen mit teurer Technologie gespickten Roboter, der Objekte erkennen und diese mit seinen Armen manipulieren kann.					
Kriterien	Autonom	x	x	-	-	Labor
techn. Funktionalität	Der Roboter kann Personen mittels (Stereo-)Kameras und 3D-Laser robust wahrnehmen und diesen durch belebte Umgebungen folgen. Der Roboter kann Türen öffnen und ist in der Lage, sich selbstständig an einer Steckdose aufzuladen.					

Pepper (Aldebaran 2008)

Einsatz	Humanoider Roboter für den häuslichen Einsatz, der darauf ausgelegt ist, Emotionen zu erkennen und den Nutzer glücklicher zu machen.					
Kriterien	Autonom	x	-	(x)	-	Wohnung
techn. Funktionalität	Vielzahl von Sensoren, um die Umgebung wahrzunehmen, Hindernissen auszuweichen und proaktiv zu handeln. Unterstützt Telekommunikation, Situationserkennung, Unterhaltungsfunktionen und Emotionserkennung (Schätzungen basierend auf emotionalen Ausdrücken und Stimmlage). Algorithmen sollen durch Cloud-AI unterstützt werden. Der Roboter kann den Nutzer per Gesichtsdetektion erkennen und sich auf Gespräche im Raum zubewegen. Weiterhin besteht eine autonome Ladefunktion.					

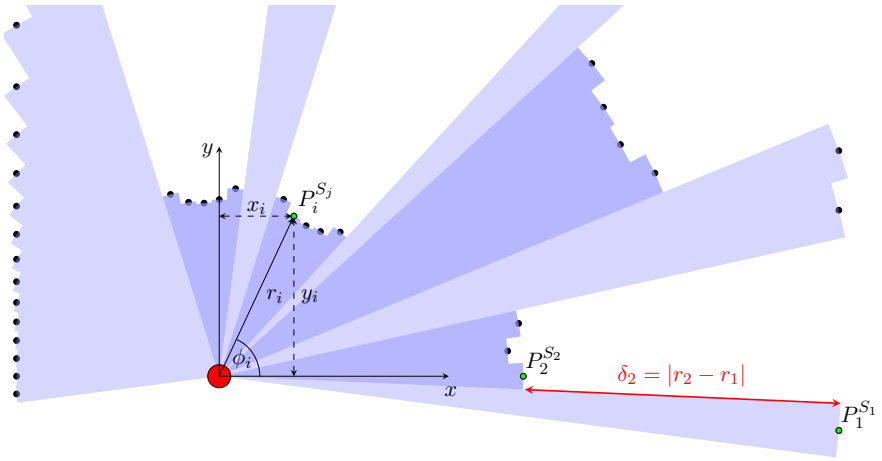


Abbildung A.2.: Ein Lasersensor liefert für jeden reflektierten Sensorkegel einen Abstandswert r_i mit zugehörigem Winkel ϕ_i. Die Sprungdistanz $d_i = |r_i - r_{i-1}|$ wird häufig verwendet, um Punkte P_i mit ähnlichem Abstandswert zu Segmenten S_j zusammenzufassen. Benachbarte Segmente sind mit unterschiedlichen Farbtönen dargestellt. Grafik in Anlehnung an Weinrich u. a. (2014b).

A.3. Detektoren

A.3.1. Aufbau eines Laserscans

Das Sensormessfeld (engl. *field of view, FOV*) eines Lasers bestimmt sich aus einem minimalen und maximalen Scanwinkel $FOV = \phi_{max} - \phi_{min}$. Innerhalb dieses Messbereichs liefert der Sensor mehrere Abstandswerte r_i mit einem zugehörigen Messwinkel ϕ_i, falls der Sensorkegel von einem Hindernis reflektiert wird. Dies ist in Abbildung A.2 verdeutlicht. Die Punkte $P_i = (r_i, \phi_i)$ an denen der Sensorkegel reflektiert wird, werden zur Weiterverarbeitung häufig von Polarkoordinaten in euklidische Koordinaten umgewandelt $P_i = (x_i, y_i)$, mit:

$$\begin{aligned} x &= r \cos\phi \\ y &= r \sin\phi \end{aligned} \tag{A.1}$$

Der in dieser Arbeit verwendete SICK Laserscanner (Abschnitt 2.2.1) besitzt ein Sensormessfeld von 270°. Die maximale messbare Entfernung beträgt ca. 15 m und die Messfrequenz 15 Hz.

A.3.2. Adaptive Boosting Algorithmus

Adaptive Boosting (AdaBoost) kombiniert mehrere schwache Klassifikatoren zu einem stärkeren Gesamtklassifikator (Freund u. a. 1997). Durch die statistische Mittelung der Klassifikatoren vermindern sich die Auswirkungen von lokalen Minima und Overfitting. Weiterhin erhöht sich die Repräsentationsfähigkeit des Gesamtklassifikators. Dabei sollten die schwachen Klassifikatoren eine möglichst hohe Diversität aufweisen, also unabhängig von den anderen Klassifikatoren zu einem Ergebnis kommen. Zur Erzeugung von Diversität werden die Klassifikatoren häufig auf unterschiedlichen Daten (Bagging, Crossvalidated Committees) trainiert oder nutzen unterschiedliche Merkmale. Beim klassischen AdaBoost (Viola u. a. 2002; Arras u. a. 2007) werden die Gewichte des Trainingsdatensatzes manipuliert und jeder schwache Klassifikator auf genau einem Merkmal trainiert. Dies führt neben einer hohen Diversität zu einer impliziten Merkmalsauswahl, indem während des Trainings der Klassifikator und damit auch das Merkmal mit dem geringsten Fehler ausgewählt wird. Algorithmus A.1 listet den AdaBoost Algorithmus als Pseudocode.

Algorithmus A.1 : AdaBoost aus Arras u. a. (2007).

Eingabe : Set von Trainingsbeispielen $D = (e_1, l_1), \ldots, (e_N, l_N)$, mit Label $l_n = +1$ für positive und $l_n = -1$ für negative Beispiele

1 Initialisiere Gewichte $w_1(n) = \frac{1}{2a}$ für $l_n = +1$ und $w_1(n) = \frac{1}{2b}$ für $l_n = -1$, mit a Anzahl der positiven und b Anzahl der negativen Beispiele

2 **for** $t = 1, \ldots, T$ **do**

3 Normalisiere Gewichte: $w_t(n) = \frac{w_t(n)}{\sum_{i=1}^{N} w_t(i)}$

4 Für jedes Merkmal f_j trainiere einen schwachen Klassifikator h_j mit D und w_t

5 // Berechne Korrelation für schwache Klassifikatoren

6 Für jeden h_j berechne $r_j = \sum_{n=1}^{N} w_t(n) l_n h_j(e_n)$, wobei $h_j(e_n) \in \{+1, -1\}$

7 // Wähle besten schwachen Klassifikator

8 Wähle h_j, der $|r_j|$ maximiert und setze $(h_t, r_t) = (h_j, r_j)$

9 // Aktualisiere Gewichte

10 $w_{t+1}(n) = w_t(n) \exp(-\alpha_t l_n h_t(e_n))$, wobei $\alpha_t = \frac{1}{2} \log(\frac{1+r_t}{1-r_t})$

Ausgabe : $H(e) = sign(F(e))$, wobei $F(e) = \sum_{t=1}^{T} \alpha_t h_t(e)$

Der Algorithmus beginnt mit der gleichverteilten Gewichtung der Trainingsbeispiele (Zeile 1). In einer Folge von $t = 1, \ldots, T$ Trainingsrunden wird sukzessive ein schwacher Klassifikator $h_t(e)$ ausgewählt, welcher die mit w_t gewichteten

Beispiele mit dem geringsten Fehler klassifiziert. Zum Training der schwachen Klassifikatoren werden pro Trainingsrunde für jede Dimension $j = 1, \ldots, M$ des Merkmalsvektors f der Schwellwert θ_j und die Parität p_j bestimmt, die den Fehler über den gewichteten Trainingsdaten minimieren (Zeile 4). Die Entscheidungsfunktion eines schwachen Klassifikators ist dabei mit Gleichung (A.2) gegeben.

$$h_j(e) = \begin{cases} +1 & \text{falls } p_j f_j(e) < p_j \theta_j \\ -1 & \text{sonst} \end{cases} \tag{A.2}$$

Von allen schwachen Klassifikatoren $h_j(e) = (f_j, p_j, \theta_j)$ wird derjenige als h_t ausgewählt, der die höchste Korrelation zu den Trainingsdaten besitzt (Zeilen 6 und 8). Über die Korrelation r_t berechnet sich auch das Gewicht α_t des schwachen Klassifikators im Ensemble (Zeilen 6 und 10). Anschließend erhalten alle in dieser Runde falsch klassifizierten Beispiele ein höheres Gewicht (Zeile 10). Die Rückgabe des Algorithmus ist das Ensemble $H(e)$, welches sich aus der gewichteten Summe der ausgewählten schwachen Klassifikatoren zusammensetzt.
Die vorgeschlagenen Entscheidungsbäume dieser Arbeit ersetzen die Entscheidungsfunktion der Klassifikatoren in Gleichung (A.2) und werden somit in Zeile 4 verwendet.

A.3.3. Entscheidungsbäume

Entscheidungsbäume sind nicht-parametrische überwachte Lernverfahren, die die Klassenzugehörigkeit eines Beispiels anhand von einfachen gelernten Entscheidungsregeln schätzen. Zum Training des Baumes werden pro Ebene die Merkmale ausgewählt, die die jeweiligen Trainingsdaten am besten in zwei Teile unterteilen. Als Kriterium wird in dieser Arbeit der Klassifikationsfehler verwendet (Gleichung (A.9)). Dies wird so lange durchgeführt, bis alle Daten einer Teilung dieselbe Klasse besitzen oder die maximale Tiefe des Baumes erreicht ist, beziehungsweise eine weitere Teilung keine zusätzlichen Informationen liefert (Wengefeld 2014). Entscheidungsbäume neigen dazu, sich mit jeder neuen Entscheidungsstufe (engl. *stump*) zu stark an die Trainingsdaten anzupassen (engl. *overfit*). Man wirkt dem entgegen, indem die maximale Tiefe der Bäume beschränkt wird, die Bäume nach dem Training ausgedünnt werden (engl. *pruning*) oder eine gewisse Anzahl von Trainingsbeispielen, die in einem Blatt fallen müssen, gefordert werden (statistische Relevanz). In Kombination mit dem AdaBoost Algorithmus sind in der Praxis oftmals schon wenige Entscheidungsknoten pro Baum ausreichend, da mehrere Bäume durch den Algorithmus kombiniert werden. Nutzt man nur eine Entscheidungsstufe und $h_t(e_n) \in \{-1, +1\}$ Ausgaben, reduziert sich das Verfahren auf die vorgestellten

eindimensionalen schwachen Klassifikatoren aus Arras u. a. (2007).

Mathematische Formulierung

Der Abschnitt listet die mathematische Formulierung eines binären Entscheidungsbaums[4] (Breiman u. a. 1984).

Training Mit gegebenen Trainingsvektor $x_i \in \mathbb{R}^n, i = 1, \ldots, l$ und Labelvektor $y \in \mathbb{R}^l$ partitioniert ein Entscheidungsbaum rekursiv den Merkmalsraum, indem Beispiele mit demselben Label gruppiert werden.
Die Daten, die an einem Knoten m anliegen, seien mit Q bezeichnet. Für jede mögliche Teilung $\theta = (j, t_m)$, wobei j das Merkmal und t_m den Schwellwert bezeichnet, werden die Daten in zwei Teilsets $Q_{left}(\theta)$ und $Q_{right}(\theta)$ partitioniert:

$$\begin{aligned} Q_{left}(\theta) &= (x, y), \text{ wobei } x_j <= t_m \\ Q_{right}(\theta) &= Q \setminus Q_{left}(\theta) \end{aligned} \tag{A.3}$$

Die Güte von Knoten m wird mit einer Kostenfunktion $H(\cdot)$ berechnet, welche von der Aufgabe abhängig ist (Klassifikation oder Regression):

$$G(Q, \theta) = \frac{n_{left}}{N_m} H(Q_{left}(\theta)) + \frac{n_{right}}{N_m} H(Q_{right}(\theta)) \tag{A.4}$$

Wähle den Parameter, der die Kostenfunktion minimiert:

$$\theta^* = \operatorname*{argmin}_{\theta} G(Q, \theta) \tag{A.5}$$

Dies wird rekursiv so lange für die entstehenden Teilsets $Q_{left}(\theta^*)$ und $Q_{right}(\theta^*)$ durchgeführt, bis das Abbruchkriterium erreicht ist. Als Abbruchkriterien kann die maximale Baumtiefe beschränkt werden oder $N_m = 1$ beziehungsweise $N_m < \min_{Beispiele}$ gefordert sein.

Klassifikationskriterium Gegeben sei eine Klassifikationsaufgabe bei der die Labels Werte von $0, 1, \ldots, K - 1$ für einen Knoten m mit Region R_m mit N_m Beispielen annehmen. Das Verhältnis von Beispielen der Klasse k in Knoten m sei mit:

$$p_{mk} = 1/N_m \sum_{x_i \in R_m} I(y_i = k) \tag{A.6}$$

[4] Erläuterungen in Anlehnung an: http://scikit-learn.org/stable/modules/tree.html

angegeben. Gebräuchliche Kostenkriterien sind der Gini Index (siehe Gleichung (A.7)), die Kreuz-Entropy (Gleichung (A.8)) und der Klassifikationsfehler (Gleichung (A.9)).

$$H(X_m) = \sum_k p_{mk}(1 - p_{mk}) \tag{A.7}$$

$$H(X_m) = \sum_k p_{mk} \log(p_{mk}) \tag{A.8}$$

$$H(X_m) = 1 - \max(p_{mk}) \tag{A.9}$$

A.3.4. Deformable Part Model

Dieser Abschnitt beschreibt das Deformable Part Model aus Abschnitt 3.3.4. Die Erläuterungen stammen aus Reuther (2011) und Laschka (2013)[5] und der ursprünglichen Veröffentlichung von P. F. Felzenszwalb u. a. (2010). Aufgrund der Komplexität des Verfahrens werden an dieser Stelle nur die wichtigsten Punkte aufgeführt. Für weiterführende Erläuterungen sei auf Reuther (2011) verwiesen.

Modell

Filter Ein Filter F ist ein rechteckiges Template aus Gewichtsvektoren. Der *Score* eines Filters wird für eine Featurekarte G und eine Position (x, y) als Skalarprodukt der Filtergewichte und des Unterfensters der Featurekarte definiert:

$$\text{score}(F) = \sum_{x',y'} F[x', y'] \cdot G[x + x', y + y'] \tag{A.10}$$

Der Filter wird auf den Skalierungsstufen einer Auflösungspyramide H angewendet. Es sei $\phi(H, p, w, h)$ die Aneinanderreihung der Merkmalsvektoren des durch $p = (x, y, l)$ definierten Unterfensters der Größe $w \cdot h$ in Ebene l der Merkmalspyramide. Analog wird mit F' die Aneinanderreihung der Merkmalsvektoren des Filters F bezeichnet. Der Score des Filters F in Bezug auf eine Position p in der Auflösungspyramide H definiert sich als:

$$\text{score}(F) = F' \cdot \phi(H, p) \tag{A.11}$$

Deformationsmodell Für jeden Part i ist eine ideale Position (Ankerpunkt) v_i definiert. Da die Part-Filter auf der doppelten Auflösung des Root-Filters

[5] Vom Autor im Rahmen dieser Arbeit betreut.

berechnet werden, ergibt sich die ideale Position a_i des Ankerpunkts bezüglich des Root-Filters zu:

$$a_i = 2(x_0, y_0) + v_i \quad \text{mit } (x_0, y_0) = \text{Position des Root-Filters} \tag{A.12}$$

Für die Beweglichkeit der Parts wird eine Kostenfunktion für eine Abweichung $(dx_i, dy_i) = (x_i, y_i) - a_i$ von der idealen Platzierung definiert. Die Kosten bestimmen sich mittels einer quadratischen Funktion über dieser Abweichung:

$$d_i \cdot \phi_d(dx_i, dy_i) \quad \text{mit } \phi_d(dx, dy) = (dx, dy, dx^2, dy^2) \tag{A.13}$$

Um dieses Modell anzupassen, müssen die Koeffizienten $d_i = (c_1, c_2, c_3, c_4)$ spezifiziert werden. Diese werden während des Trainings gelernt.

Detektion

Um die Antwort der Filter des Modells zu berechnen, muss zunächst die optimale Positionierung der Parts einer Hypothese bestimmt werden.

Berechnung des Scores Sei $z = (p_0, \ldots, p_n)$ eine Hypothese, wobei $p_i = (x_i, y_i, l_i)$ die Position des i-ten Filters in der Ebene l der Merkmalspyramide H bestimmt. Der Score des Filters sei gegeben mit:

$$\text{score}(z) = \sum_{i=0}^{n} F_i' \cdot \phi\left(H, p_i\right) - \sum_{i=1}^{n} d_i \cdot \phi_d\left(dx_i, dy_i\right) + b \tag{A.14}$$

Er ergibt sich als Summe aller Filterantworten an den jeweiligen Positionen, abzüglich der Deformationskosten für die Abweichung der Parts von ihrer idealen Position. Die Scores verschiedener Komponenten können aufgrund der unterschiedlichen Filter stark voneinander abweichen. Daher wird für jede Komponente ein Bias b gelernt, um sie mit anderen Komponenten vergleichbar zu machen. Die Deformationskosten $(dx_i, dy_i) = (x_i, y_i) - (2(x_0, y_0) + v_i)$ des i-ten Parts bezüglich seiner Ankerposition v_i sind durch eine quadratische Funktion $\phi_d(dx, dy)$ definiert (Gleichung (A.13)).

Bestimmung der optimalen Part-Konfiguration Für die Anwendung von Formel A.14 wird angenommen, dass eine vollständige Hypothese in der Form $z = (p_0, ..., p_n)$ vorliegt. Dies bedeutet, dass für eine Root-Filter-Position p_0 eine Part-Filter-Konfiguration $(p_1, ..., p_n)$ vorliegt. Zu jeder Position des Root-Filters wird die optimale Part Konfiguration gewünscht welche die Scores, abzüglich der Deformationskosten maximiert.

Zur Vermeidung von redundanten Berechnungen wird zu Beginn der Score für jeden Filter F_i in jeder Ebene der Featurepyramide H an jeder Position (x, y) berechnet und in einem Array R gespeichert:

$$R_{i,l}(x, y) = F_i' \cdot \phi(H, (x, y, l)) \tag{A.15}$$

Weiterhin wird ein Array D eingeführt, in dem für jede Position p_i der optimale Score des Part-Filters F_i berechnet wird, wenn der Part seinen Ankerpunkt v_i an Position p_i hätte:

$$D_{i,l}(x, y) = \max_{dx,dy} \Big(R_{i,l}(x + dx, y + dy) - d_i \cdot \phi_d(dx, dy) \Big) \tag{A.16}$$

Die Berechnung des Arrays D erfolgt dabei mittels *generalized distance transform* in $\mathcal{O}(nk)$, wobei n die Anzahl der Parts und k die Anzahl aller Positionen in H beschreiben. Der Score eines Modells lässt sich somit durch die Nutzung der beiden Arrays bestimmen:

$$\text{score}(x_0, y_0, l_0) = R_{0,l_0}(x_0, y_0) + \sum_{i=1}^{n} D_{i,l_0-\lambda}(2(x_0, y_0) + v_i) + b \tag{A.17}$$

Für weitere für die Detektion nötige Schritte, wie die *Bounding-Box Prediction* und die *Non-Maximum Suppression*, sowie den relativ komplizierten Trainingsalgorithmus sei auf P. F. Felzenszwalb u. a. (2010) verwiesen.

A.4. Kameraprojektion

A.4.1. Lochkameraprojektion

Dieser Abschnitt skizziert den Ablauf der Lochkameraprojektion, welche in dieser Arbeit für die Kameras des Roboters angenommen wird[6]. Bei der Projektion wird der Weltpunkt $P_k = (x_k, y_k, z_k)$ in Kamerakoordinaten auf den Bildpunkt $P_i = (u, v)$ projiziert.

[6] Das Bild des verwendeten Fischaugenobjektivs lässt sich mittels Kamerakalibrierung in ein Lochkameramodell überführen: https://sites.google.com/site/scarabotix/ocamcalib-toolbox.

Die Projektion kann mittels homogener Koordinaten beschrieben werden:

$$P_i' = \mathbf{K}[\mathbf{I}\ \ \mathbf{0}]P_k \tag{A.18}$$

$$\begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_k \\ y_k \\ z_k \\ 1 \end{pmatrix}$$

wobei die intrinsische Kameramatrix $\mathbf{K}$ mit den Brennweiten f_x, f_y in Pixeln und dem Bildmittelpunkt (c_x, c_y) gebildet wird. Der Punkt $P_i' = (u', v', w')^T$ wird anschließend normiert:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} u'/w' \\ v'/w' \\ w'/w' \end{pmatrix} \Rightarrow \begin{pmatrix} u \\ v \end{pmatrix} \tag{A.19}$$

um den Bildpunkt $P_i = (u, v)^T$ zu erhalten[7]. Die Tiefeninformation, also die Distanz zum Objekt, geht dabei verloren, sodass bei der Rückprojektion nur ein Strahl bestimmt werden kann, auf dem der Weltpunkt liegt.

A.4.2. Inverse Lochkameraprojektion

Die inverse Lochkameraprojektion ermittelt aus einem Bildpunkt P_i einen Punkt P_k' im Kamerakoordinatensystem:

$$P_k' = \mathbf{K}^{-1} P_i \tag{A.20}$$

$$\begin{pmatrix} x_k' \\ y_k' \\ z_k' \end{pmatrix} = \begin{pmatrix} \frac{1}{f_x} & 0 & -\frac{c_x}{f_x} \\ 0 & \frac{1}{f_y} & -\frac{c_y}{f_y} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{A.21}$$

Da die Tiefe w nicht mehr rekonstruiert werden kann, ergibt sich nur ein Strahl vom Kamerazentrum zum Pixel auf der Bildebene. Wird $w = 1$ gesetzt ergibt sich $z_k' = 1$. Über Kenntnis der Breite des Objekts in der Welt und der Rechteckbreite im Bild kann die tatsächliche Entfernung des Objekts berechnet werden. Die Strahllänge vom Kamerazentrum zum Weltpunkt ergibt sich zu:

$$s = \frac{b_k f_x}{b_i} \tag{A.22}$$

wobei b_i die Breite der Bounding-Box, f_x die Brennweite in x und b_k die Ob-

[7] Im Allgemeinen wird zusätzlich noch eine Korrektur der Linsenverzeichnung vorgenommen.

jektbreite in der Welt bezeichnen. Der Punkt in Kamerakoordinaten ergibt sich demnach zu:

$$P_k = sP'_k \tag{A.23}$$

A.5. Bayes-Filter Algorithmen

Dieser Abschnitt beschreibt die mathematischen und algorithmischen Details der in der Arbeit verwendeten Bayes-Filter.

A.5.1. Kalman-Filter Algorithmus

Algorithmus A.2 : Kalman-Filter (vgl. Thrun u. a. (2005))

Eingabe : $\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1}, \mathbf{u}_t, \mathbf{z}_t$

1 // Prädiktionsschritt

2 $\bar{\boldsymbol{\mu}}_t = \mathbf{A}_t \boldsymbol{\mu}_{t-1} + \mathbf{B}_t \mathbf{u}_t$

3 $\bar{\boldsymbol{\Sigma}}_\mathbf{t} = \mathbf{A}_t \boldsymbol{\Sigma}_{t-1} \mathbf{A}_t^T + \mathbf{R}_t$

4 // Korrekturschritt

5 $\mathbf{K}_t = \bar{\boldsymbol{\Sigma}}_t \mathbf{C}_t^T \left(\mathbf{C}_t \bar{\boldsymbol{\Sigma}}_t \mathbf{C}_t^T + \mathbf{Q}_t \right)^{-1}$

6 $\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + \mathbf{K}_t \left(\mathbf{z}_t - \mathbf{C}_t \bar{\boldsymbol{\mu}}_t \right)$

7 $\boldsymbol{\Sigma}_t = \left(\mathbf{I} - \mathbf{K}_t \mathbf{C}_t \right) \bar{\boldsymbol{\Sigma}}_\mathbf{t}$

Ausgabe : $\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t$

Die Eingabe des Algorithmus umfasst den zuletzt geschätzten Zustand $\boldsymbol{\mu}_{t-1}$, die zugehörige Kovarianz $\boldsymbol{\Sigma}_{t-1}$, die Systemsteuerung $\mathbf{u}_t$ sowie die aktuelle Messung $\mathbf{z}_t$.
Im Prädiktionsschritt wird der prädizierte Belief $\overline{bel}(\mathbf{x}_t)$, durch $\bar{\boldsymbol{\mu}}_t$ und $\bar{\boldsymbol{\Sigma}}_t$ repräsentiert, berechnet. Während sich $\bar{\boldsymbol{\mu}}_t$ aus dem deterministischen Teil von Gleichung (4.1) ergibt, berücksichtigt die Aktualisierung der Kovarianz die Tatsache, dass der neue Zustand aus dem vorherigen Zustand über die lineare Matrix $\mathbf{A}_t$ abhängt. $\mathbf{R}_t$ bezeichnet die additive gaußsche Rauschmatrix.
Im Korrekturschritt wird der Belief $\overline{bel}(\mathbf{x}_t)$ durch die Messung $\mathbf{z}_t$ zum Belief $bel(\mathbf{x}_t)$ transformiert. Die Matrix $\mathbf{K}_t$ bezeichnet dabei den sogenannten Kalman Gain. Sie bestimmt zu welchem Anteil die Beobachtung die neue Zustandsschätzung beeinflusst. Anschließend wird der Mittelwert proportional zum Kalman Gain und der Abweichung von der aktuellen Beobachtung $\mathbf{z}_t$ zur prädizierten Beobachtung $\mathbf{C}_t \bar{\boldsymbol{\mu}}_t$ (Gleichung (4.3)) angepasst. Die Gewichtung

des Kalman Gains kann folgendermaßen erklärt werden. Angenommen die Dimension der Beobachtung entspricht dem Zustand, also $\mathbf{C}_t = \mathbf{I}$, dann folgt $\mathbf{K}_t = \bar{\mathbf{\Sigma}}_t \left(\bar{\mathbf{\Sigma}}_t + \mathbf{Q}_t\right)^{-1}$. Demnach entspricht der Kalman Gain dem Verhältnis zwischen der Kovarianz des prädizierten Zustands und der Beobachtung. Bei einem sicheren prädizierten Zustand und einer verrauschten Beobachtung wird $\mathbf{K}_t$ klein, umgekehrt groß. Die Differenz aus prädizierter und aktueller Beobachtung $(\mathbf{z}_t - \mathbf{C}_t\bar{\boldsymbol{\mu}}_t)$ wird auch als Innovation bezeichnet. Der Informationsgewinn der Beobachtung wird zuletzt dafür genutzt, um die Kovarianz $\mathbf{\Sigma}_t$ des neuen Beliefs zu berechnen.
Die Ausgabe des Algorithmus umfasst den neuen Erwartungswert $\boldsymbol{\mu}_t$ und die Kovarianz $\mathbf{\Sigma}_t$ des Systemzustands. Für die mathematische Herleitung des Kalman-Filters sei auf Thrun u. a. (2005, Seite 45 ff.) verwiesen.

A.5.2. Extended Kalman-Filter

Der Extended Kalman-Filter beschreibt die Wahrscheinlichkeit der Zustandstransition und die Beobachtungswahrscheinlichkeit durch die nichtlinearen Funktionen g und h:

$$\begin{aligned} \mathbf{x}_t &= g(\mathbf{u}_t, \mathbf{x}_{t-1}) + \boldsymbol{\varepsilon}_t \\ \mathbf{z}_t &= h(\mathbf{x}_t) + \boldsymbol{\delta}_t \end{aligned} \tag{A.24}$$

Demnach ersetzt die Funktion $g(\cdot)$ die Matrizen $\mathbf{A}_t$ und $\mathbf{B}_t$ aus der Gleichung (4.1) und Funktion $h(\cdot)$ ersetzt Matrix $\mathbf{C}_t$ aus Gleichung (4.3). Nichtlineare Funktionen für g und h zerstören die Normalverteilung des Beliefs. Der EKF linearisiert daher g und h mittels Taylorreihe durch lineare Funktionen, die tangential zu einem bestimmten Funktionswert verlaufen. Der Anstieg der Tangente bestimmt sich mittels der partiellen Ableitung:

$$\mathbf{g}'(\mathbf{u}_t, \mathbf{x}_{t-1}) := \frac{\partial \mathbf{g}(\mathbf{u}_t, \mathbf{x}_{t-1})}{\partial \mathbf{x}_{t-1}} \tag{A.25}$$

Als Stützstelle der Linearisierung bietet es sich an, den Erwartungswert der letzten Schätzung $\boldsymbol{\mu}_{t-1}$ zu verwenden, da dieser den wahrscheinlichsten Zustand repräsentiert (Thrun u. a. 2005). g wird demnach durch die Taylorreihe im Funktionswert bei $\boldsymbol{\mu}_{t-1}$ und $\mathbf{u}_t$ approximiert:

$$\begin{aligned} g(\mathbf{u}_t, \mathbf{x}_{t-1}) &\approx g(\mathbf{u}_t, \boldsymbol{\mu}_{t-1}) + \underbrace{g'(\mathbf{u}_t, \boldsymbol{\mu}_{t-1})}_{=:\mathbf{G}_t}(\mathbf{x}_{t-1} - \boldsymbol{\mu}_{t-1}) \\ &= g(\mathbf{u}_t, \boldsymbol{\mu}_{t-1}) + \mathbf{G}_t(\mathbf{x}_{t-1} - \boldsymbol{\mu}_{t-1}) \end{aligned} \tag{A.26}$$

Die Matrix $\mathbf{G}$ wird als Jacobi-Matrix bezeichnet. Analog dazu wird im Beobachtungsupdate der neue geschätzte Systemzustand $\bar{\boldsymbol{\mu}}_\mathbf{t}$ als Linearisierungspunkt verwendet:

$$\begin{aligned} h(\mathbf{x}_t) &\approx h(\bar{\boldsymbol{\mu}}_t) + \underbrace{h'(\bar{\boldsymbol{\mu}}_t)}_{=:\mathbf{H}_t}(\mathbf{x}_t - \bar{\boldsymbol{\mu}}_t) \\ &= h(\bar{\boldsymbol{\mu}}_t) + \mathbf{H}_t(\mathbf{x}_t - \bar{\boldsymbol{\mu}}_t) \end{aligned} \tag{A.27}$$

wobei $h'(\mathbf{x}_t) = \frac{\partial h(\mathbf{x}_t)}{\partial \mathbf{x}_t}$.

Algorithmus A.3 : Extended Kalman-Filter (vgl. Thrun u. a. (2005))

Eingabe : $\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1}, \mathbf{u}_t, \mathbf{z}_t$

1 // Prädiktionsschritt
2 $\bar{\boldsymbol{\mu}}_t = g(\mathbf{u}_t, \boldsymbol{\mu}_{t-1})$
3 $\bar{\boldsymbol{\Sigma}}_\mathbf{t} = \mathbf{G}_t \boldsymbol{\Sigma}_{t-1} \mathbf{G}_t^T + \mathbf{R}_t$
4 // Korrekturschritt
5 $\mathbf{K}_t = \bar{\boldsymbol{\Sigma}}_t \mathbf{H}_t^T \left(\mathbf{H}_t \bar{\boldsymbol{\Sigma}}_t \mathbf{H}_t^T + \mathbf{Q}_t \right)^{-1}$
6 $\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + \mathbf{K}_t \left(\mathbf{z}_t - h(\bar{\boldsymbol{\mu}}_t) \right)$
7 $\boldsymbol{\Sigma}_t = \left(\mathbf{I} - \mathbf{K}_t \mathbf{H}_t \right) \bar{\boldsymbol{\Sigma}}_\mathbf{t}$

Ausgabe : $\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t$

Wie man in Algorithmus A.3 erkennen kann, ist die Berechnung ähnlich zum Kalman-Filter aus Algorithmus A.2. In Zeile 2 wird jedoch der neue Zustand nicht mehr über die Matrizen $\mathbf{A}_t$ und $\mathbf{B}_t$ geschätzt, sondern durch die nichtlineare Funktion $g(\mathbf{u}_t, \boldsymbol{\mu}_{t-1})$. Genauso verhält es sich bei der Berechnung von $\boldsymbol{\mu}_t$. Hier wird $h(\bar{\boldsymbol{\mu}}_t)$ verwendet. Für die Kovarianzmatrix und den Kalman Gain kommen die linearisierten Matrizen $\mathbf{G}_\mathbf{t}$ und $\mathbf{H}_\mathbf{t}$ zur Anwendung (Thrun u. a. 2005).

A.5.3. Unscented Kalman-Filter

Beim UKF erfolgt die Linearisierung über sogenannte Sigma Punkte $\mathcal{X}^{[i]}$. Diese approximieren die nichtlineare Funktion über die gesamte Breite der Gaußverteilung. Hierzu werden Sigma Punkte aus der Normalverteilung extrahiert und durch $g(\cdot)$ transformiert. Die Sigma Punkte befinden sich auf dem Mittelwert und symmetrisch entlang jeder Richtung der Kovarianz. Für eine n-dimensionale Gaußverteilung werden die $2n + 1$ Sigma Punkte $\mathcal{X}^{[i]}$ wie folgt

bestimmt:

$$\begin{aligned} \mathcal{X}^{[0]} &= \boldsymbol{\mu} && \text{(A.28)} \\ \mathcal{X}^{[i]} &= \boldsymbol{\mu} + \left(\sqrt{(n+\lambda)\boldsymbol{\Sigma}}\right)_i && \text{für } i = 1, \ldots, n \\ \mathcal{X}^{[i]} &= \boldsymbol{\mu} - \left(\sqrt{(n+\lambda)\boldsymbol{\Sigma}}\right)_{i-n} && \text{für } i = n+1, \ldots, 2n \end{aligned}$$

Dabei ist $\lambda = \alpha^2(n+\kappa) - n$ wobei α und κ bestimmen, wie weit die Sigma Punkt vom Mittelwert entfernt liegen. Jeder Sigma Punkt hat zwei Gewichte:

$$\begin{aligned} w_m^{[0]} &= \frac{\lambda}{n+\lambda} && \text{(A.29)} \\ w_c^{[0]} &= \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta) \\ w_m^{[i]} &= w_c^{[i]} = \frac{1}{2(n+\lambda)} && \text{für } i = 1, \ldots, 2n \end{aligned}$$

$w_m^{[i]}$ wird dabei zur Berechnung des Mittelwerts verwendet. Das Gewicht $w_c^{[i]}$ dient der Wiederherstellung der Kovarianz. β kann genutzt werden, um zusätzliches Wissen über, die der Gaußrepräsentation zugrunde liegende Verteilung zu codieren. Im Falle einer tatsächlichen Gaußverteilung ist $\beta = 2$ optimal. Die Sigmapunkte werden anschließend mit der nichtlinearen Funktion $g(\cdot)$ verrechnet.

$$\mathcal{Y}^{[i]} = g(\mathcal{X}^{[i]}) \qquad \text{(A.30)}$$

Der neue Mittelwert $\boldsymbol{\mu'}$ und die neue Kovarianz $\boldsymbol{\Sigma'}$ werden aus den transformierten Sigma Punkten $\mathcal{Y}^{[i]}$ extrahiert

$$\begin{aligned} \boldsymbol{\mu'} &= \sum_{i=0}^{2n} w_m^{[i]} \mathcal{Y}^{[i]} && \text{(A.31)} \\ \boldsymbol{\Sigma'} &= \sum_{i=0}^{2n} w_c^{[i]} \left(\mathcal{Y}^{[i]} - \boldsymbol{\mu'}\right)\left(\mathcal{Y}^{[i]} - \boldsymbol{\mu'}\right)^T \end{aligned}$$

In Algorithmus A.4 ist der Algorithmus als Pseudocode gegeben (Thrun u. a. 2005). In Zeile 2 werden die Sigma Punkte aus dem vorherigen Belief berechnet mit $\gamma = \sqrt{n+\lambda}$. Diese Punkte werden in Zeile 3 mittels $g(\cdot)$ prädiziert. Anschließend werden der prädizierte Mittelwert und die Kovarianz berechnet (Zeilen 4 und 5). In Zeile 5 wird ebenfalls das Prädiktionsrauschen $\mathbf{R}_t$ hinzugefügt. Daraufhin wird in Zeile 7 ein neues Set von Sigma Punkten aus der Gaußverteilung extrahiert. In Zeile 9 wird für jeden Sigma Punkt eine Beob-

Algorithmus A.4 : Unscented Kalman-Filter (vgl. Thrun u. a. (2005))

Eingabe : $\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1}, \mathbf{u}_t, \mathbf{z}_t$

// Prädiktion

$\mathcal{X}_{t-1} = (\boldsymbol{\mu}_{t-1} \quad \boldsymbol{\mu}_{t-1} + \gamma\sqrt{\boldsymbol{\Sigma}_{t-1}} \quad \boldsymbol{\mu}_{t-1} - \gamma\sqrt{\boldsymbol{\Sigma}_{t-1}})$ // Sigmapunkte

$\bar{\mathcal{X}}_t^* = g(\mathbf{u}_t, \mathcal{X}_{t-1})$ // Prädiktion der Sigmapunkte über g

$\bar{\boldsymbol{\mu}}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{X}}_t^{*[i]}$

$\bar{\boldsymbol{\Sigma}} = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{X}}_t^{*[i]} - \bar{\boldsymbol{\mu}}_t)(\bar{\mathcal{X}}_t^{*[i]} - \bar{\boldsymbol{\mu}}_t)^T + \mathbf{R}_t$

// Beobachtungsupdate

$\bar{\mathcal{X}}_t = (\bar{\boldsymbol{\mu}}_t \quad \bar{\boldsymbol{\mu}}_t + \gamma\sqrt{\bar{\boldsymbol{\Sigma}}_t} \quad \bar{\boldsymbol{\mu}}_t - \gamma\sqrt{\bar{\boldsymbol{\Sigma}}_t})$ // Berechne Sigmapunkte aus Prädiktion

$\bar{\mathcal{Z}}_t = h(\bar{\mathcal{X}}_t)$ // Transformation zur geschätzten Beobachtung

$\hat{\mathbf{z}}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{Z}}_t^{[i]}$ // prädizierten Beobachtung

$\mathbf{S}_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{Z}}_t^{[i]} - \hat{\mathbf{z}}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{\mathbf{z}}_t)^T + \mathbf{Q}_t$ // prädizierte Kovarianzmatrix der Beobachtung

$\bar{\boldsymbol{\Sigma}}_t^{x,z} = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{X}}_t^{[i]} - \bar{\boldsymbol{\mu}}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{\mathbf{z}}_t)^T$ // Kreuzkovarianz zw. Prädiktion und prädizierter Beobachtung

$\mathbf{K}_t = \bar{\boldsymbol{\Sigma}}_t^{x,z} \mathbf{S}_t^{-1}$ // Kalman Gain

$\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + \mathbf{K}_t(\mathbf{z}_t - \hat{\mathbf{z}}_t)$

$\boldsymbol{\Sigma}_t = \bar{\boldsymbol{\Sigma}}_t - \mathbf{K}_t \mathbf{S}_t \mathbf{K}_t^T$

Ausgabe : $\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t$

achtung $\mathcal{Z}_t$ prädiziert welche genutzt wird, um die prädizierte Beobachtung $\hat{\mathbf{z}}_t$ und deren Kovarianz $\mathbf{S}_t$ zu berechnen. Matrix $\mathbf{Q}_t$ beschreibt das additive Messrauschen. Zeile 11 bestimmt die Kreuzkovarianz zwischen dem Zustand und der prädizierten Beobachtung. Daraufhin wird der Kalman Gain berechnet und der neue Mittelwert und die Kovarianz bestimmt.

A.6. Systemmodelle

A.6.1. Konstante Geschwindigkeit mit zufälliger Beschleunigung

Systemzustand

Als Zustandsrepräsentation wird eine 6-dimensionale Gaußverteilung verwendet, die Position und Geschwindigkeit von Personen erfasst.

$$\mathbf{x}_t = (x_t, y_t, z_t, \dot{x}_t, \dot{y}_t, \dot{z}_t) \tag{A.32}$$

Bewegungsmodell

Das lineare Bewegungsmodell in Zeile 1 aus Algorithmus A.2 definiert sich mit den Matrizen:

$$\mathbf{A}_t = \begin{pmatrix} 1 & 0 & 0 & \triangle t & 0 & 0 \\ 0 & 1 & 0 & 0 & \triangle t & 0 \\ 0 & 0 & 1 & 0 & 0 & \triangle t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{R}_t = \gamma^2 \begin{pmatrix} \frac{\triangle t^4}{4} & 0 & 0 & \frac{\triangle t^3}{2} & 0 & 0 \\ 0 & \frac{\triangle t^4}{4} & 0 & 0 & \frac{\triangle t^3}{2} & 0 \\ 0 & 0 & \frac{\triangle t^4}{4} & 0 & 0 & \frac{\triangle t^3}{2} \\ \frac{\triangle t^3}{2} & 0 & 0 & \triangle t^2 & 0 & 0 \\ 0 & \frac{\triangle t^3}{2} & 0 & 0 & \triangle t^2 & 0 \\ 0 & 0 & \frac{\triangle t^3}{2} & 0 & 0 & \triangle t^2 \end{pmatrix} \tag{A.33}$$

Durch Matrix $\mathbf{A}_t$ ergibt sich das Modell der konstanten Geschwindigkeit. Die unbekannte Beschleunigung wird als zufällig angenommen und im Rauschterm $\mathbf{R}_t$ modelliert. $\mathbf{R}_t$ ergibt sich dabei mit:

$$\mathbf{R}_t = \mathbf{G}\ diag(\mathbf{q})\ \mathbf{G}^T \tag{A.34}$$

$$\mathbf{q} = \begin{pmatrix} \sigma_{a_x}^2 \\ \sigma_{a_y}^2 \\ \sigma_{a_z}^2 \end{pmatrix} \quad \mathbf{G} = \begin{pmatrix} \frac{\triangle t^2}{2} & 0 & 0 \\ 0 & \frac{\triangle t^2}{2} & 0 \\ 0 & 0 & \frac{\triangle t^2}{2} \\ \triangle t & 0 & 0 \\ 0 & \triangle t & 0 \\ 0 & 0 & \triangle t \end{pmatrix}$$

In Gleichung (A.33) wurde $\gamma^2 = \sigma_{a_x}^2 = \sigma_{a_y}^2 = \sigma_{a_z}^2$ angenommen. Damit ist die Unsicherheit der Beschleunigung in alle Raumrichtungen gleich. Matrix $\mathbf{G}_t$ ergibt sich nach den allgemeinen Bewegungsgleichungen, welche das Verhalten eines physikalischen Systems bezüglich dessen Bewegung als Funktion der Zeit beschreiben (Lerner u. a. 1991):

$$\begin{aligned} s &= s_0 + v_0 t + \frac{a}{2} t^2 \\ v &= v_0 + at \end{aligned} \tag{A.35}$$

Hier entspricht s_0 der initialen Position, s der finalen Position, v_0 der initialen und v der finalen Geschwindigkeit und a der Beschleunigung. Das Rauschen ist damit von der vergangenen Zeit abhängig.

Beobachtungsmodell

Als Beobachtung wird eine Gaußverteilung über die Position der Person im Raum verwendet.

$$\mathbf{z}_t = (x'_t, y'_t, z'_t) \tag{A.36}$$

Damit ergibt sich für das Beobachtungsmodell in Zeile 4 aus Algorithmus A.2 mit:

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad \mathbf{Q}_t = \mathbf{\Sigma}_{Beobachtung} \tag{A.37}$$

Die Matrix $\mathbf{Q}_t$ entspricht dabei der in Weltkoordinaten transformierten (Gleichung (3.7)), fehlerpropagierten (Gleichung (3.13)) und ausgerichteten (Abschnitt 3.5.6) Kovarianzmatrix der Beobachtung $\hat{\mathbf{\Sigma}}_k$ nach Gleichung (3.5):

$$\hat{\mathbf{\Sigma}}_k = \begin{pmatrix} \sigma^2_{x_k} & 0 & 0 \\ 0 & \sigma^2_{y_k} & 0 \\ 0 & 0 & \sigma^2_{z_k} \end{pmatrix} \tag{A.38}$$

Die Geschwindigkeiten der Raumrichtungen werden demnach nicht beobachtet, sondern durch den Kalman-Filter geschätzt.

A.6.2. Konstante Orientierung und Geschwindigkeit

Systemzustand

Das nichtlineare, 5-dimensionale Modell nach Bellotto u. a. (2009) verwendet eine Gaußverteilung, welche die Position, die Orientierung in der x-y Ebene und die Geschwindigkeit entlang der Orientierung beinhaltet:

$$\mathbf{x}_t = (x_t, y_t, z_t, \phi_t, v_t) \tag{A.39}$$

Bewegungsmodell

Für das Bewegungsmodell aus Zeile 1 in Algorithmus A.3 bzw. Zeile 1 in Algorithmus A.4 ergibt sich folgender funktionaler Zusammenhang des Systemzustands:

$$\bar{\mathbf{x}}_t = \begin{cases} x_t = x_{t-1} + v_{t-1} \triangle t \ \cos\phi_{t-1} \\ y_t = y_{t-1} + v_{t-1} \triangle t \ \sin\phi_{t-1} \\ z_t = z_{t-1} + \varepsilon^z_{t-1} \\ \phi_t = \phi_{t-1} + \varepsilon^\phi_{t-1} \\ v_t = |v_{t-1}| + \varepsilon^v_{t-1} \end{cases} \tag{A.40}$$

ε beschreibt einen Gaußverteilten Rauschterm, der auf z, ϕ und v addiert wird. Die Jacobi-Matrix ergibt sich durch partielle Ableitung jeder Zeile von $g(\mathbf{x})$ nach jeder Variable:

$$\mathbf{G}_t = \begin{pmatrix} 1 & 0 & 0 & -v_{t-1} \triangle t \sin\phi_{t-1} & \triangle t \cos\phi_{t-1} \\ 0 & 1 & 0 & v_{t-1} \triangle t \cos\phi_{t-1} & \triangle t \sin\phi_{t-1} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & sgn(v_{t-1}) \end{pmatrix} \tag{A.41}$$

Dabei ist $sgn(x)$ definiert mit:

$$sgn(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases} \tag{A.42}$$

Der Rauschterm $\mathbf{R}_t$ ergibt sich nach Gleichung (A.34) aus dem Vektor $\mathbf{q}$ und der Matrix $\mathbf{G}$:

$$\mathbf{q} = \begin{pmatrix} \sigma_z^2 \\ \sigma_\phi^2 \\ \sigma_v^2 \end{pmatrix} \quad \mathbf{G} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{A.43}$$

Mit diesen Zusammenhängen ist das Bewegungsmodell für den EKF und UKF definiert.

Beobachtungsmodell

Als Beobachtung wird analog zu Anhang A.6.1 die Position der Person mit zugehöriger Kovarianz in Weltkoordinaten verwendet.

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad \mathbf{Q}_t = \mathbf{\Sigma}_{Beobachtung} \tag{A.44}$$

Somit werden die Orientierung ϕ_t und die Geschwindigkeit v_t gefiltert.

A.7. Datensätze zur Evaluation des Personentrackings

Dieser Abschnitt beschreibt die zur Evaluation verwendeten Datensätze. Alle Personen in den Datensätzen sind händisch mit Bounding-Boxen, IDs und Verdeckungsinformation mithilfe des VATIC Label-Tools annotiert (Carl Vondrick 2012).

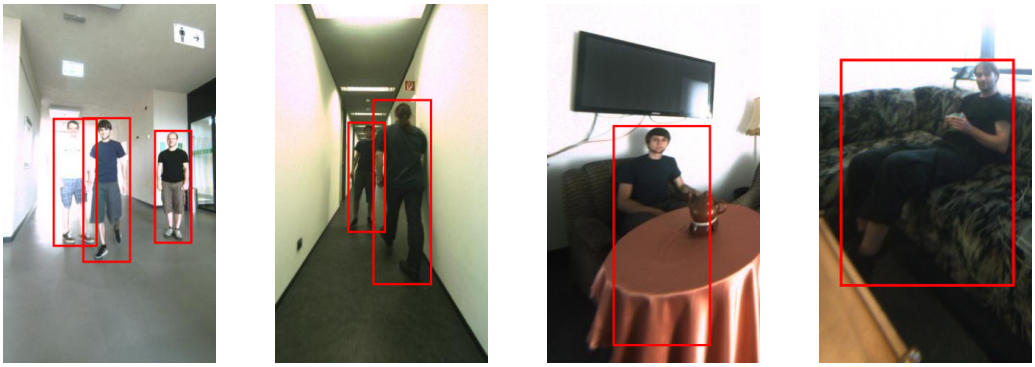
A.7.1. Datensätze aus Volkhardt u. a. (2013a)

Der Datensatz umfasst 8 verschiedene Testszenarien, die mit dem Roboter aufgenommen wurden und alle nötigen Sensordaten (z. B. Laserdaten, Bilder, Roboterpose, Umgebungskarte) enthalten[8]. Die Umgebung umfasst eine, in einem Labor, nachgestellte Wohnung (siehe Abbildung A.5(a)). Im Folgenden werden die Datensätze der Übersichtlichkeit nach ihrem Inhalt in jeweils 4

[8] Online verfügbar unter: `http://www.tu-ilmenau.de/neurob/data-sets-code/people-tracking-in-home-environments/`

Tabelle A.1.: Statistiken der Datensätze.

Datensatz	Länge	Frames	Inhalt
Stehend	46 s	621	1-4 laufende Personen
Folgen	110 s	1578	Folgen 1-2 Personen
Sitzend	82 s	1310	1 sich setzende Person
Suche	217 s	2909	1-2 sitzende Personen



(a) Stehend **(b)** Folgen **(c)** Sitzend **(d)** Suche

Abbildung A.3.: Beispielhafte gelabelte Bilder aus den Datensätzen. (a) Stehender Roboter mit mehreren Personen, (b) folgender Roboter mit kreuzender Person, (c) Roboter mit sich setzender und aufstehender Person, (d) suchender Roboter mit sitzenden und stehenden Personen.

Gruppen zusammengefasst. Tabelle A.1 und Abbildung A.3 geben einen Eindruck über den Inhalt und die Schwierigkeit der einzelnen Datensätze.

- **Stehend:** Der erste Datensatz enthält einen statischen Roboter und bis zu 4 Personen, die sich frei vor dem Roboter bewegen. Dabei kommt es zu häufigen Verdeckungen.
- **Folgen:** Im Datensatz folgt der Roboter dem Nutzer und eine andere Person kreuzt den Pfad des Roboters.
- **Sitzend:** Der Datensatz beinhaltet einen stehenden Roboter mit einer Person, die das Sichtfeld des Roboters betritt, sich hinsetzt und anschließend wieder aufsteht und die Szene verlässt.
- **Suche:** Im Datensatz sucht der Roboter sitzende Nutzer in der Wohnung. Gelegentlich stehen diese auf, sobald sie sich im Sichtbereich befinden, und setzen sich erneut.

Tabelle A.2.: Statistiken der Datensätze.

Datensatz	Länge	Frames	Inhalt
Dataset 1	103 s	1556	3 Personen vor Roboter
Dataset 2	91 s	1375	Nutzerfolgen mit kreuzender Person
Dataset 3-8	1133 s	15973	Suche mit unterschiedlichen Nutzern und Beleuchtungsbedingungen

A.7.2. Datensätze mit erhöhter Schwierigkeit

Die Datensätze mit erhöhter Schwierigkeit wurden in der gleichen nachgestellten Wohnumgebung wie die Datensätze aus Anhang A.7.1 aufgenommen (siehe Abbildung A.5(a)). Um die Szenarien realistischer zu machen, wurde die Wohnung mit zusätzlichen häuslichen Gegenständen und Möbeln ausgestattet. Weiterhin wurde die Laborbeleuchtung aus Abbildung A.3 durch Steh- und Hängelampen ersetzt und Rollos angebracht, um Aufnahmen bei Dämmerung und Nacht zu simulieren. Tabelle A.2 listet Statistiken und den Inhalt der Datensätze. Einen visuellen Eindruck der Datensätze liefert Abbildung A.4.

- **Dataset 1:** umfasst einen überwiegend stehenden Roboter und 3 Personen, die sich frei vorm Roboter bewegen. Dabei gehen die Personen verschiedenen Tätigkeiten nach.
- **Dataset 2:** enthält ein Folgenszenario, bei dem der Roboter einem Nutzer durch die gesamte Wohnung folgt und 1 andere Person die Wege des Roboters kreuzt.
- **Dataset 3-8:** Die Datensätze umfassen mehrere Suchszenarien.

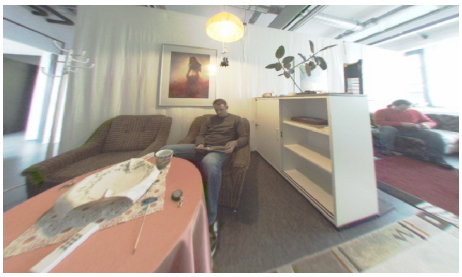
Wie in Abbildungen A.4(c) bis A.4(h) sichtbar umfassen die Szenarien unterschiedliche Posen und Tätigkeiten der Nutzer, z. B. das Lesen eines Buches. Weiterhin wurde auf realistische Beleuchtungsbedingungen geachtet. So enthält der Datensatz in Abbildung A.4(g) nur wenig Licht von einer Deckenlampe und ausgeprägte Schatten. In Abbildung A.4(f) wurde durch zugezogene Vorhänge und fehlende zusätzliche Lichtquellen eine dunkle Wohnung bei Nacht simuliert.

A.8. Testumgebungen

Die in der Arbeit verwendeten Testumgebungen umfassen ein nachgestelltes häusliches Szenario und reale Wohnungen von Projektmitarbeiten und Senio-

(a) Dataset 1

(b) Dataset 2



(c) Dataset 3

(d) Dataset 4

(e) Dataset 5

(f) Dataset 6

(g) Dataset 7

(h) Dataset 8

Abbildung A.4.: Datensätze mit erhöhter Schwierigkeit. Die Datensätze enthalten viele häusliche Einrichtungsgegenstände, natürliche Nutzerposen und realistische Beleuchtungsbedingungen.

ren. Abbildung A.5 stellt die Grundrisse und vorherrschenden Beleuchtungsbedingungen der Wohnungen dar. Die nachgestellte Wohnung im Labor aus Abbildung A.5(a) umfasst einen Flur, eine Küche, einen Gästeraum und ein Wohnzimmer (vgl. auch Abbildung 5.4(b)). Die Testumgebung wurde für die Erzeugung der Datensätze aus Anhang A.7 verwendet und diente damit zur Evaluation der Detektoren und des Personentrackers. Auch die Experimente zur Personensuche und zur Sturzdetektion wurden in der Testumgebung durchgeführt.
Abbildungen A.5(b) bis A.5(h) zeigen einen Auszug aus den Grundrissen der verwendeten realen Wohnungen von Mitarbeitern und Senioren. In diesen wurden der Personentracker, das Folgen einer Person und die Personensuche untersucht. Auch wurden hier die mehrtägigen Experimente aus Kapitel 7 durchgeführt.

Einfluss der Beleuchtungsbedingungen

Für die Personenerkennung ist auch die Beleuchtungssituation der Wohnung zu berücksichtigen, da ein Großteil der Detektoren visuell arbeitet. Ein hoher Kontrastumfang der Szenen kann für die Personenwahrnehmung Probleme bereiten, wenn der Dynamikumfang der Kamera nicht ausreicht. In allen Testwohnungen herrschen unterschiedliche Beleuchtungsbedingungen vor. Abbildung A.5 zeigt Bereiche, in denen kaum Tageslicht durch die Fenster einfällt, in Blau, Regionen mit Tageslicht sind rot eingefärbt. Fensternahe Bereiche sind im Allgemeinen heller, können aber bei direkter Sonneneinstrahlung eine Überbelichtung der Kamera verursachen. In den Bereichen ohne Fenster sind in den Seniorenwohnungen meist nur wenige künstliche Lichtquellen vorhanden. Die geringe Helligkeit verursacht entweder dunkle Bilder oder erzeugt im Automatikmodus der Kamera lange Belichtungszeiten und ein vermehrtes Bildrauschen. Eine lange Belichtungszeit führt vor allem bei Bewegungen des Roboters und der Nutzer zu Problemen. Diese Probleme machen die Notwendigkeit von multi-modalen Sensoren zur Personendetektion deutlich.

A.9. Evaluationsmetriken

Dieser Abschnitt beschreibt die, in der Arbeit verwendeten, Evaluationsmetriken.

A.9.1. ROC-Kurven

Die Receiver-Operation-Characteristic-Kurve kann eingesetzt werden, um visuell den bestmöglichen Wert eines Parameter eines binären Klassifikators zu

(a) Labor **(b)** WhgM1

(c) WhgM2 **(d)** WhgM3

(e) WhgS1 **(f)** WhgS2

(g) WhgS3 **(h)** WhgS4

Abbildung A.5.: Grundrisse der Testwohnungen mit Einrichtungsgegenständen und Fenstern. Eingefärbt sind die vom Roboter befahrenen Räume. Rot helle Bereiche mit Tageslicht und Blau dunkle Bereiche, welche nur künstliche Beleuchtung enthalten.

finden. Die ROC-Kurve wird gebildet, indem für verschiedene Arbeitspunkte (Schwellwert über der Ausgabe) oder Parameter eines Klassifikators die Richtig-positiv-Rate mit der Falsch-positiv-Rate ins Verhältnis gesetzt wird. Es ergibt sich das typische Bild aus Abbildung 3.3. Zur besseren Vergleichbarkeit verschiedener Kurven berechnet man entweder die Fläche unter der Kurve (engl. *area under curve*) oder nutzt die geringste Balanced Error Rate (BER) über alle Arbeitspunkte. Die BER berechnet sich nach:

$$BER = \frac{1}{2}\left(\frac{FN}{P} + \frac{FP}{N}\right) \tag{A.45}$$

Die geringste Balanced Error Rate entspricht dem nordwestlichsten Punkt der Kurve und ist somit der Schwellwert bzw. Parameterwert, ab dem bei einer weiteren Erhöhung mehr falsche Klassifikationen als richtige hinzukommen würden (Wengefeld 2014).

A.9.2. $\mathcal{F}$-Score

Der $\mathcal{F}$-Score kombiniert *Precision* und *Recall* mittels gewichtetem harmonischen Mittel (Rijsbergen 1979):

$$\mathcal{F} = \frac{2 \cdot PR \cdot RC}{PR + RC} \,. \tag{A.46}$$

Das entspricht die Precision der Anzahl der richtig-positiven Klassifikationen durch die Anzahl aller positiven Klassifikationen (also die vom Klassifikator zurückgelieferten Richtig- und Falsch-Positiven). Der Recall bezeichnet der richtig-positiv Klassifikationen durch die Anzahl aller positiven Beispiele. Der $\mathcal{F}_\alpha$-score kann genutzt werden, um jeweils Precision oder Recall ein höheres Gewicht zu geben:

$$\mathcal{F}_\alpha = \frac{(1+\alpha^2) \cdot PR \cdot RC}{\alpha^2 \cdot PR + RC} \,. \tag{A.47}$$

So bewertet der $\mathcal{F}_2$-score den Recall 4-mal so stark wie die Precision, während der $\mathcal{F}_{0.5}$-score die Precision 4-mal so stark gewichtet, wie den Recall.

A.9.3. Intersection-over-Union

Die *Intersection-over-Union* wird genutzt, um die Ähnlichkeit zweier Bounding-Boxen zu bestimmen. Sie kann genutzt werden, um Detektionen oder Hypothesen in Form von Bounding-Boxen mit einer gelabelten Ground-Truth zu vergleichen:

$$IoU = \frac{A \cap B}{A \cup B} \,. \tag{A.48}$$

Die Intersection-over-Union berechnet sich also aus der Fläche der Überschneidung der Bounding-Boxen A und B durch die Vereinigung der beiden Boxen. Alternative, ähnliche Distanzmaße zur Bewertung von Bounding-Box Detektionen finden sich in Agarwal u. a. (2004) und Leibe (2005).

A.9.4. Multiple Object Tracking Performance (MOT)

Die MOT (Bernardin u. a. 2008) berechnet die mittleren *Misses* ($\overline{\text{Miss}}$), die mittleren *Falsch-positiven* ($\overline{\text{FP}}$), den mittleren *Missmatch error* ($\overline{\text{MME}}$), die *multiple object tracking precision* (MOTP) und die *multiple object tracking accuracy* (MOTA). In der Arbeit werden häufig noch *Recall* (RC) und *Precision* (PR) angegeben.
Die ersten 3 Werte beschreiben jeweils das Verhältnis aus den summierten Misses (Falsch-negativen), den Falsch-positiven und den Missmatches (Verwechslungen) und der Gesamtanzahl an *Ground-Truth* Objekten. Die MOTP beschreibt den mittleren Fehler in der geschätzten Position für alle assoziierten Hypothese-Ground-Truth Paare. Die Distanz von Detektion und Hypothese wird im Falle von Bounding-Boxen mit der *Intersection-over-union* (Anhang A.9.3) und bei Posen mit der euklidischen Distanz bestimmt. Die Genauigkeit und Konsistenz des Trackers ist mit dem wichtigsten Maß der Metrik, der MOTA angegeben:

$$MOTA = 1 - \frac{\sum_k (Miss_k + FP_k + MME_k)}{\sum_k G_k} \,, \tag{A.49}$$

wobei $Miss_k$ die Misses, FP_k die Falsch-positiven, MME_k die Missmatches und G_k die Anzahl aller gelabelten Objekte zum Zeitschritt k beschreiben. Bei der MOTA bedeutet ein Wert von 1 ein perfektes Tracking ohne fehlende Objekte, keine Falsch-positiven und keine ID-Verwechslungen (konsistente Tracks). Nach unten ist die MOTA nicht beschränkt und kann leicht negativ werden, vor allem im Falle vieler falsch-positiv Detektionen.

A.10. Evaluation des Personentrackings

A.10.1. Evaluation aus Volkhardt u. a. (2013a)

Dieser Abschnitt fasst die Ergebnisse aus Volkhardt u. a. (2013a) zusammen.

Auswertungsmethodik und Bewertungsmetriken

Zur Evaluation werden alle 3D Hypothesen des Trackers zu Bounding-Boxen im Kamerabild transformiert (Anhang A.4.2). Die obere Position der Bounding-Box ergibt sich aus der Höhe der Hypothese, während für die untere Position angenommen wird, dass Personen den Boden berühren. Die Breite der Box wird empirisch auf die halbe Höhe gesetzt.
Die Bounding-Boxen und ihre IDs werden mit den gelabelten Bounding-Boxen mithilfe der *Multiple Object Tracking Performance* bewertet (Anhang A.9.4). Diese berechnet die Präzision, Genauigkeit und Konsistenz des Trackers. Als Distanzmaß wird die *Intersection-over-Union*-Metrik verwendet (siehe Anhang A.9.3). Ist der Intersection-over-Union Abstand zwischen Hypothese und Ground-Truth geringer als ein Schwellwert von 0.25 wird eine richtig-positiv Detektion gezählt. Bei Mehrfach-Detektionen auf einem Ground-Truth-Label wird nur eine Detektion als richtig-positiv gewertet, während alle Weiteren als Falsch-positive zählen.
Gelabelte Bounding-Boxen, die zu mehr als 50 % verdeckt sind, werden besonders ausgewertet und alle Interaktionen mit diesen ignoriert. Das heißt einerseits, dass fehlende Detektionen auf verdeckten Bounding-Boxen nicht die Richtig-positiv-Rate vermindern. Dies wird erreicht, indem die verdeckten Ground-Truth Boxen nicht zur Gesamtheit der positiven Beispiele hinzugezählt werden. Andererseits zählt eine Detektion auf einer verdeckten Box auch nicht zur Menge der Richtig-positiven. Weiterhin werden Mehrfach-Detektionen auf der gleichen verdeckten Bounding-Box nicht als Falsch-positive gewertet. Im Falle der MOT müssen die IDs der Hypothesen auch nach der Verdeckung weiterhin korrekt sein, andernfalls wird ein Missmatch Error gezählt.

Ergebnisse

Nachfolgend werden die Ergebnisse der Evaluation auf den Datensätzen aus Anhang A.7.1 ausgeführt. Tabelle A.3 vergleicht den Personentracker im Echtzeitbetrieb unter Nutzung des Gesichts-, HOG-, Oberkörper-HOG-, Bewegungs- und Beinpaardetektors mit einem Tracker, welcher nur Beinpaardetektionen nutzt. Die Ergebnisse eines Personentrackers, welcher jedes Kamerabild offline mit einem FPDW-Detektor verarbeitet und einer Kombination aus FPDW- und Beinpaardetektor sind in Tabelle A.4 zu finden. Tabelle A.5 listet analog die Ergebnisse eines offline DPM-Detektors[9]. *Precision* und *Recall* der einzelnen Detektoren sind in Tabelle A.6 angegeben.

[9]Zum Zeitpunkt der Veröffentlichung lagen der FPDW- und der DPM-Detektor noch nicht in echtzeitfähiger Version auf dem Roboter vor (siehe Abschnitt 3.3)

Tabelle A.3.: Vergleich von Personen- und Beinpaartracker

Datensatz	$\overline{\text{Miss}}$	$\overline{\text{FP}}$	$\overline{\text{MME}}$	RC	PR	MOTP	MOTA
Stehend	0.30	0.28	0.0109	0.76	0.73	0.50	0.40
Nur Beinpaar	0.40	0.24	0.0100	0.66	0.73	0.51	0.35
Folgen	0.26	0.28	0.0122	0.77	0.73	0.51	0.45
Nur Beinpaar	0.24	0.39	0.0071	0.79	0.67	0.52	0.35
Sitzend	0.43	0.55	0.0066	0.59	0.51	0.54	0.02
Nur Beinpaar	0.49	0.19	0.0102	0.52	0.74	0.53	0.32
Suche	0.51	0.48	0.0044	0.49	0.52	0.61	0.01
Nur Beinpaar	0.55	0.87	0.0094	0.45	0.44	0.63	-0.43

Tabelle A.4.: Vergleich von FPDW- und FPDW+Beinpaartracker

Datensatz	$\overline{\text{Miss}}$	$\overline{\text{FP}}$	$\overline{\text{MME}}$	RC	PR	MOTP	MOTA
Stehend	0.51	0.34	0.0075	0.50	0.59	0.55	0.14
+ Beinpaar	0.28	0.40	0.0174	0.77	0.66	0.56	0.29
Folgen	0.40	0.22	0.0032	0.60	0.73	0.51	0.37
+ Beinpaar	0.19	0.31	0.0032	0.82	0.72	0.53	0.48
Sitzend	0.84	0.44	0.0065	0.17	0.27	0.64	-0.28
+ Beinpaar	0.66	0.45	0.0093	0.35	0.43	0.57	-0.11
Suche	0.94	0.40	0.0033	0.06	0.10	0.72	-0.34
+ Beinpaar	0.67	0.37	0.0058	0.33	0.54	0.55	-0.04

Tabelle A.5.: Vergleich von DPM- und DPM+Beinpaartracker

Datensatz	$\overline{\text{Miss}}$	$\overline{\text{FP}}$	$\overline{\text{MME}}$	RC	PR	MOTP	MOTA
Stehend	0.42	0.16	0.0100	0.60	0.79	0.49	0.41
+ Beinpaar	0.30	0.31	0.0125	0.74	0.70	0.49	0.37
Folgen	0.28	0.16	0.0045	0.73	0.82	0.48	0.56
+ Beinpaar	0.11	0.35	0.0045	0.95	0.73	0.49	0.54
Sitzend	0.46	0.44	0.0047	0.55	0.56	0.56	0.10
+ Beinpaar	0.36	0.51	0.0093	0.65	0.56	0.56	0.14
Suche	0.52	0.36	0.0032	0.49	0.54	0.60	0.12
+ Beinpaar	0.39	0.39	0.0033	0.61	0.62	0.57	0.22

Tabelle A.6.: Recall und Precision der Detektoren (offline auf jedem Frame)

Datensatz	FPDW		DPM	
	RC	PR	RC	PR
Stehend	0.76	0.97	0.58	0.50
Folgen	0.53	0.98	0.62	0.86
Sitzend	0.21	0.81	0.58	0.75
Suche	0.13	0.37	0.53	0.65

Der echtzeitfähige Personentracker zeigt gute Resultate, wenn Personen stehen oder laufen. Bei sitzenden Personen nimmt die Performance jedoch stark ab (Tabelle A.3). Dennoch ist die Fusion mehrerer Detektionsmodule meist einem rein Beinpaar-basierten Tracker überlegen. Im Falle des *Sitzend* Datensatz treten mehr Falsch-positive auf, die durch HOG Detektionen auf einer Stehlampe produziert werden. Die Datensätze mit sitzenden Personen zeigen, dass die verwendeten Detektoren nicht ausreichend sind, um sitzende Posen robust zu tracken. Der Personentracker hat Schwierigkeiten ruhig sitzende Personen zu erkennen, wenn deren Gesicht nicht sichtbar ist und der Oberkörper-HOG nicht anschlägt. Weiterhin weisen die *Sitzend* und *Suche* Datensätze höhere Falsch-positive auf, die meist durch Schränke, Tische, Stühle oder Stehlampen verursacht werden.

Der offline FPDW-basierte Tracker zeigt gute Resultate, wenn sich Personen in aufrechter Pose befinden (Tabelle A.4). Da der Detektor speziell für Fußgänger im Straßenverkehr trainiert wurde, verschlechtert sich die Performance des Trackers stark, wenn Personen sitzen. In allen Datensätzen kann die Performance verbessert werden, indem zusätzlich Beinpaardetektionen im Personentracker fusioniert werden. Der FPDW-basierte Tracker zeigt im Vergleich zum echtzeitfähigen Personentracker schlechtere Performance. Für stehende Personen liefert der Detektor jedoch eine hohe Precision und ausreichenden Recall (Tabelle A.6).

Die besten Resultate werden mit einem DPM-basierten Personentracker erzielt (Tabelle A.5). Die hohen Recall und Precision Werte des DPM Detektors (Tabelle A.6) führen zu den höchsten MOTA Werten in nahezu allen Datensätzen. Ein zusätzlicher Recall-Gewinn lässt sich durch die Kombination mit einem Beinpaardetektor erreichen. Allerdings führt dies zu einem Absinken der Precision und MOTA, da der Beinpaardetektor viele Falsch-positive liefert.

A.10.2. Evaluation des Personentrackers

Dieser Abschnitt enthält die erreichten Evaluationsmetriken für den Personentracker in Kombination mit unterschiedlichen Detektoren, welche in Abschnitt 4.10.4 ausgewertet wurden. Tabelle A.7 listet die MOTA Metriken, bestehend aus den mittleren Misses, den mittleren Falsch-positiven, dem mittleren Missmatch error, Recall (RC), Precision (PR), der multiple object tracking precision (MOTP) und der multiple object tracking accuracy (MOTA) sowie den $\mathcal{F}$-score (siehe Anhang A.9). Mit dem DPM- und Beinpaardetektor werden die besten Ergebnisse erzielt. Daher wurde diese Kombination für das reale Einsatzszenario ausgewählt.

Tabelle A.7.: Evaluationsmetriken für Personentracker mit verschiedenen Detektoren gruppiert nach dem Inhalt der Datensätze

Detektoren / Datensatz	$\overline{\text{Miss}}$	$\overline{\text{FP}}$	$\overline{\text{MME}}$	RC	PR	MOTP	MOTA	$\mathcal{F}_{score}$
DPM								
Stehend	0.31	0.14	0.0046	0.69	0.82	0.25	0.55	0.75
Folgend	0.19	0.17	0.0035	0.81	0.83	0.22	0.63	0.82
Sitzend	0.54	0.49	0.0019	0.46	0.48	0.26	-0.04	0.47
Suche	0.54	0.31	0.0047	0.46	0.62	0.27	0.15	0.51
HOGs								
Stehend	0.35	0.21	0.0047	0.65	0.75	0.26	0.44	0.70
Folgend	0.22	0.25	0.0035	0.78	0.76	0.22	0.53	0.77
Sitzend	0.50	0.25	0.0034	0.50	0.66	0.25	0.24	0.55
Suche	0.56	0.44	0.0057	0.44	0.57	0.24	0.00	0.47
DPM+FPDW								
Stehend	0.29	0.24	0.0065	0.71	0.74	0.23	0.47	0.72
Folgend	0.14	0.22	0.0042	0.86	0.79	0.20	0.63	0.82
Sitzend	0.55	0.67	0.0043	0.45	0.41	0.27	-0.22	0.42
Suche	0.56	0.45	0.0054	0.44	0.52	0.28	-0.02	0.46
HOGs+FPDW								
Stehend	0.23	0.52	0.0070	0.77	0.59	0.24	0.24	0.67
Folgend	0.12	0.49	0.0049	0.88	0.64	0.20	0.39	0.74
Sitzend	0.59	0.57	0.0043	0.41	0.46	0.26	-0.17	0.42
Suche	0.53	0.75	0.0074	0.47	0.44	0.26	-0.28	0.43

A.11. Explorative Suche

A.11.1. Partikelschwarm Optimierung

Die Partikelschwarm Optimierung ist ein stochastisches Verfahren, dass eine Menge von Partikeln verwendet, um den Zustandsraum abzudecken. Jedes Partikel bewegt sich nach einem vorgegebenen Modell im Zustandsraum und wird anschließend mittels einer Kostenfunktion bewertet. Ein Partikel i ist durch 3 Eigenschaften definiert:

1. Zustand $\mathbf{x}_i$
2. Geschwindigkeit $\mathbf{v}_i$
3. lokales Optimum $\mathbf{p}_i$

Zusätzlich wird das globale Optimum $\hat{\mathbf{x}}$ aller Partikel gespeichert. Die Partikel erhalten in jedem Optimierungsschritt eine Beschleunigung in Richtung ihres lokalen Optimums als auch auf das globale Optimum. Dadurch bewegen sich die Partikel während der Optimierung in ihrem lokalen Bereich, mit zunehmenden Iterationen jedoch schwarmartig auf das globale Optimum zu. Die einzelnen Verarbeitungsschritte sind in Algorithmus A.5 beschrieben.

Dabei bezeichnen $\mathbf{b}_{min}$ und $\mathbf{b}_{max}$ das Minimum und Maximum des Suchraums. Die weiteren Parameter werden in der Praxis, wie auch in dieser Arbeit, meist zu $\omega = 0.7$, $\phi_p = 1.4$ und $\phi_g = 1.4$ gesetzt. Die Anzahl der Partikel beträgt im Fall der explorativen Suche 500.

A.11.2. Aktualisierung der Aufenthaltswahrscheinlichkeit

Dieser Abschnitt beschreibt den binären Bayes-Filter zur Aktualisierung der Aufenthaltswahrscheinlichkeitskarte.

In Algorithmus A.6 beschreibt $l_{t,i}$ das *log odds* Verhältnis von $p(\mathbf{m}_i|z_{1:t})$. l_0 entspricht dem Prior jeder Zelle in log odds Form bevor Sensormessungen getätigt werden. Der Updatebereich umfasst die Bereiche der Sensorkegel, die vor einem Hindernis im Freiraum der Occupancy-Karte liegen. Der Belief von Zellen, die nicht im Updatebereich liegen, das heißt außerhalb des Sensorkegels oder in bzw. hinter Hindernissen, wird nicht verändert. Der mittlere Term in Zeile 3 bezeichnet das inverse Sensormodell.

Algorithmus A.5 : Partikelschwarm Optimierung (Clerc 2012)

Eingabe : S // Partikelset

foreach *Partikel* $i = 1, \ldots, S$ **do**
  $\mathbf{x}_i \sim \mathcal{U}(\mathbf{b}_{min}, \mathbf{b}_{max})$ // Initialisiere Position gleichverteilt
  $\mathbf{p}_i \leftarrow \mathbf{x}_i$ // Initialisiere lokales Optimum
  if $f(\mathbf{p}_i) < f(\mathbf{g})$ **then**
    $\mathbf{g} \leftarrow \mathbf{p}_i$ // Aktualisiere das globale Optimum
  $\mathbf{v}_i \sim \mathcal{U}(-|\mathbf{b}_{max} - \mathbf{b}_{min}|, |\mathbf{b}_{max} - \mathbf{b}_{min}|)$ // Initialisiere Geschwindigkeit
  while *Terminierungskriterium nicht erfüllt* **do**
    foreach *Partikel* $i = 1, \ldots, S$ **do**
      foreach *Dimension* $d = 1, \ldots, n$ **do**
        $r_p, r_g \sim \mathcal{U}(0, 1)$ // Generiere Zufallszahlen
        $\mathbf{v}_{i,d} \leftarrow \omega \mathbf{v}_{i,d} + \phi_p r_p(\mathbf{p}_{i,d} - \mathbf{x}_{i,d}) + \phi_g r_g(\mathbf{g}_d - \mathbf{x}_{i,d})$
        // Aktualisiere Geschwindigkeit
        $\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i$ // Aktualisiere Position
        if $f(\mathbf{x}_i) < f(\mathbf{p}_i)$ **then**
          $\mathbf{p}_i \leftarrow \mathbf{x}_i$ // Aktualisiere lokales Optimum
          if $f(\mathbf{p}_i) < f(\mathbf{g})$ **then**
            $\mathbf{g} \leftarrow \mathbf{p}_i$ // Aktualisiere globales Optimum

Ausgabe : $\hat{S}$ // aktualisiertes Partikelset

Algorithmus A.6 : Aktualisierung der Aufenthaltswahrscheinlichkeitskarte (Vgl. Thrun u. a. (2005))

Eingabe : $m = \{\mathbf{m}_i\}$ mit $\{l_{t-1,i}\}, x_t, z_t$ // Aufenthaltskarte, Roboterpose, Beobachtung

foreach *Zellen* $\mathbf{m}_i$ **do**
  if $\mathbf{m}_i$ *in Updatebereich von* z_t **then**
    $l_{t,i} = l_{t-1,i} + log\frac{p(\mathbf{m}_i|z_t,x_t)}{1-p(\mathbf{m}_i|z_t,x_t)} - l_0$
  else
    $l_{t,i} = l_{t-1,i}$

Ausgabe : $m = \{\mathbf{m}_i\}$ mit $\{l_{t,i}\}$ // aktualisierte Aufenthaltskarte

Inverses Sensormodell

Das inverse Sensormodell $p(\mathbf{m}_i|z_t, x_t)$ in *log odds* Form ist gegeben mit:

$$log\frac{p(\mathbf{m}_i|z_t, x_t)}{1 - p(\mathbf{m}_i|z_t, x_t)} \tag{A.50}$$

Als einfaches Modell wird im vorliegenden Fall $p(\mathbf{m}_i|z_t, x_t) = 0.8$ gesetzt, wenn sich die Zelle $\mathbf{m}_i$ in einem festgelegten Radius von $1\,m$ um eine detektierte Hypothese befindet. Falls sich die Zelle außerhalb des Radius befindet, wird $p(\mathbf{m}_i|z_t, x_t) = 0.45$ gesetzt. Dies bewirkt, dass sich die Aufenthaltswahrscheinlichkeit mit jeder Beobachtung verringert, falls keine Person erkannt wurde. Auf der anderen Seite erhöht sich die Aufenthaltswahrscheinlichkeit von Zellen im Radius um eine Hypothese mit jeder Beobachtung.

Abbildungsverzeichnis

Literaturverzeichnis

Aeberhard, M., S. Paul, N. Kaempchen und T. Bertram (Juni 2011). "Object existence probability fusion using dempster-shafer theory in a high-level sensor data fusion architecture". In: *2011 IEEE Intelligent Vehicles Symposium (IV)*, S. 770–775 (siehe S. 70).

Agarwal, S., A. Awan und D. Roth (Nov. 2004). "Learning to detect objects in images via a sparse, part-based representation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.11, S. 1475–1490 (siehe S. 158).

Aldebaran (2008). *Nao und Pepper*. `http://www.aldebaran.com/en` (siehe S. 17, 134).

Alfred (2014). *Webseite*. `http://alfred.eu/` (siehe S. 17).

ALIAS (2010). *Webseite*. `www.aal-alias.eu` (siehe S. 17).

Altendorfer, R. und S. Matzka (Juni 2010). "A confidence measure for vehicle tracking based on a generalization of Bayes estimation". In: *2010 IEEE Intelligent Vehicles Symposium*. Bd. 1. IEEE, S. 766–772 (siehe S. 70).

Amigoni, F. und V. Caglioti (2010). "An information-based exploration strategy for environment mapping with mobile robots". In: *Robotics and Autonomous Systems* 58.5, S. 684–699 (siehe S. 97).

Anderson, D., J. Keller, M. Skubic, X. Chen und Z. H. (2006). "Recognizing Falls from Silhouettes". In: *Int. Conf. on Engineering in Medicine and Biology Society*, S. 6388–6391 (siehe S. 111).

Andriluka, M., S. Roth und B. Schiele (Juni 2008). "People-tracking-by-detection and people-detection-by-tracking". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, S. 1–8 (siehe S. 54).

Arenknecht, R. (2015). "Implementierung und Evaluation verschiedener Bayesfilter für das Personentracking." Bachlorarbeit. TU Ilmenau (siehe S. 63, 80).

Arfken, G. und H. Weber (2008). *Mathematical methods for physicists*. Elsevier Acad. Press (siehe S. 45).

Arnaud Doucet, N. d. F. Ñ. G., Hrsg. (2001). *Sequential Monte Carlo Methods in Practice*. Springer (siehe S. 62).

Arras, K. O., S. Grzonka, M. Luber und W. Burgard (Mai 2008). "Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities". In: *2008 IEEE International Conference on Robotics and Automation*. IEEE, S. 1710–1715 (siehe S. 55).

Arras, K. O., O. M. Mozos und W. Burgard (Apr. 2007). "Using Boosted Features for the Detection of People in 2D Range Data". In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. April. IEEE, S. 3402–3407 (siehe S. 26 f., 29 f., 40, 51, 118, 136, 138).

Arulampalam, M., S. Maskell, N. Gordon und T. Clapp (Feb. 2002). "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking". In: *Signal Processing, IEEE Transactions on* 50.2, S. 174–188 (siehe S. 62).

Bajracharya, M., B. Moghaddam, A. Howard, S. Brennan und L. H. Matthies (Juli 2009). "A Fast Stereo-based System for Detecting and Tracking Pedestrians from a Moving Vehicle". In: *The International Journal of Robotics Research* 28.11-12, S. 1466–1485 (siehe S. 20, 54).

Bar-Shalom, Y. (Juli 2002). "Update with out-of-sequence measurements in tracking: exact solution". In: *IEEE Transactions on Aerospace and Electronic Systems* 38.3, S. 769–777 (siehe S. 69).

Bar-Shalom, Y., T. Kirubarajan und X.-R. Li (2002). *Estimation with Applications to Tracking and Navigation*. New York, NY, USA: John Wiley & Sons, Inc. (siehe S. 66).

Bar-Shalom, Y. und X. R. Li (1996). *Multitarget-Multisensor Tracking: Principles and Techniques.* Bd. 16. 1, S. 93 (siehe S. 66, 70).

Barker, A. L., D. E. Brown und W. N. Martin (1994). "Bayesian Estimation and the Kalman Filter". In: *Computers Math. Applic* 30, S. 55–77 (siehe S. 58).

Basso, F., M. Munaro, S. Michieletto, E. Pagello und E. Menegatti (2013). "Fast and Robust Multi-people Tracking from RGB-D Data for a Mobile Robot". English. In: *Intelligent Autonomous Systems 12*. Hrsg. von S. Lee, H. Cho, K.-J. Yoon und J. Lee. Bd. 193. Advances in Intelligent Systems and Computing. Springer Berlin Heidelberg, S. 265–276 (siehe S. 56).

Bauml, B., F. Schmidt, T. Wimbock, O. Birbach, A. Dietrich, M. Fuchs u. a. (Mai 2011). "Catching flying balls and preparing coffee: Humanoid Rollin'Justin performs dynamic and sensitive tasks". In: *Robotics and Automa-*

tion (ICRA), 2011 IEEE International Conference on, S. 3443–3444 (siehe S. 17).

Bellotto, N. und H. Hu (Aug. 2007). "People Tracking and Identification with a Mobile Robot". In: *2007 International Conference on Mechatronics and Automation*. IEEE, S. 3565–3570 (siehe S. 66).

– (Dez. 2009). "Computationally efficient solutions for tracking people with a mobile robot: an experimental evaluation of Bayesian filters". In: *Autonomous Robots* 28.4, S. 425–438 (siehe S. 20, 55, 60, 63, 80, 150).

– (2010). "A Bank of Unscented Kalman Filters for Multimodal Human Perception with Mobile Service Robots". In: *International Journal of Social Robotics* 2.2, S. 121–136 (siehe S. 55).

Benenson, R., M. Mathias, R. Timofte und L. Van Gool (2012). "Pedestrian detection at 100 frames per second". In: *CVPR* (siehe S. 32, 51).

Benenson, R., M. Omran, J. Hosang und B. Schiele (2014). "Ten years of pedestrian detection, what have we learned?" In: *ECCV, CVRSUAD workshop* (siehe S. 32).

Bernardin, K. und R. Stiefelhagen (2008). "Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics". In: *EURASIP Journal on Image and Video Processing* 2008, S. 1–10 (siehe S. 158).

Bhattacharyya, A. (1943). "On A Measure of Divergence Between Two Statistical Populations Defined by their Probability Distributions". In: *Bulletin of Cal. Math. Soc.* 35.1, S. 99–109 (siehe S. 86).

Borges, G. A. und M.-J. Aldon (2004). "Line Extraction in 2D Range Images for Mobile Robotics". In: *Journal of Intelligent and Robotic Systems* 40.3, S. 267–297 (siehe S. 26).

Borie, R., C. Tovey und S. Koenig (2011). "Algorithms and complexity results for graph-based pursuit evasion". In: *Autonomous Robots* 31.4, S. 317 (siehe S. 97).

Bourdev, L. und J. Brandt (2005). "Robust object detection via soft cascade". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Bd. 2, S. 236–243 (siehe S. 38).

Breiman, L., J. Friedman, R. Olshen und C. Stone (1984). *Classification and Regression Trees*. Monterey, CA: Wadsworth und Brooks (siehe S. 27, 138).

Breitenstein, M., F. Reichlin, B. Leibe, E. Koller-Meier und L. Van Gool (Sep. 2011). "Online Multiperson Tracking-by-Detection from a Single, Unca-

librated Camera". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33.9, S. 1820–1833 (siehe S. 55).

Brell, M., T. Frenken, J. Meyer und A. Hein (2010). "A Mobile Robot for Self-selected Gait Velocity Assessments in Assistive Environments". In: *Proc. 3rd Intl. Conf. on Pervasive Technologies Related to Assistive Environments (PETRA'10)*. Samos, Greece (siehe S. 17).

Brèthes, L., F. Lerasle, P. Danès und M. Fontmarty (Okt. 2008). "Particle filtering strategies for data fusion dedicated to visual tracking from a mobile robot". In: *Machine Vision and Applications* 21.4, S. 427–448 (siehe S. 56).

Carballo, A., A. Ohya und S. Yuta (Aug. 2008). "Fusion of double layered multiple laser range finders for people detection from a mobile robot". In: *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*, S. 677–682 (siehe S. 26).

Care-O-Bot 4 (2015). *Webseite*. `http://www.care-o-bot-4.de/` (siehe S. 17).

Carl Vondrick Donald Patterson, D. R. (2012). "Efficiently Scaling Up Crowdsourced Video Annotation". In: *IJCV* (siehe S. 74, 151).

Chen, L., P. O. Arambel und R. K. Mehra (2002). "Estimation under unknown correlation: covariance intersection revisited". In: *IEEE Trans. Automat. Contr.* 47.11, S. 1879–1882 (siehe S. 67).

Cho, H., P. E. Rybski, A. Bar-Hillel und W. Zhang (Juni 2012). "Real-time pedestrian detection with deformable part models". In: *2012 IEEE Intelligent Vehicles Symposium*, S. 1035–1042 (siehe S. 37).

Choi, W., C. Pantofaru und S. Savarese (Nov. 2013). "A general framework for tracking multiple people from a moving camera." In: *IEEE transactions on pattern analysis and machine intelligence* 35.7, S. 1–15 (siehe S. 21, 51, 55).

Chung, T. H., G. A. Hollinger und V. Isler (Nov. 2011). "Search and Pursuit-evasion in Mobile Robotics". In: *Autonomous Robots* 31.4, S. 299–316 (siehe S. 97).

Cielniak, G., T. Duckett und A. J. Lilienthal (Mai 2010). "Data association and occlusion handling for vision-based people tracking by mobile robots". In: *Robotics and Autonomous Systems* 58.5, S. 435–443 (siehe S. 15, 56).

Clerc, M. (Sep. 2012). "Standard Particle Swarm Optimisation". 15 pages (siehe S. 164).

Cogniron (2004). *Webseite*. `www.cogniron.org` (siehe S. 16).

CompanionAble (2008). *Integrated Cognitive Assistive & Domotic Companion Robotic Systems for Ability & Security (EU-FP7 project).* Webseite: `www.companionable.net` (siehe S. 2, 16).

Cosgun, A., D. A. Florencio und H. I. Christensen (2013). "Autonomous person following for telepresence robots". In: *Proceedings - IEEE International Conference on Robotics and Automation*, S. 4335–4342 (siehe S. 82).

Cox, I. J. (1993). "A Review of Statistical Data Association Techniques for Motion Correspondence". In: *International Journal of Computer Vision* 10, S. 53–66 (siehe S. 67).

Craig, J. J. (2005). *Introduction to Robotics: Mechanics and Control].* 3. Aufl. Prentice Hall (siehe S. 48).

Dalal, N. und B. Triggs (2005). "Histograms of oriented gradients for human detection". In: *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'05).* IEEE, S. 886–893 (siehe S. 31–34, 40, 51).

Dietmayer, K., A. Kirchner und N. Kämpchen (2005). "Fusionsarchitekturen zur Umfeldwahrnehmung für zukünftige Fahrerassistenzsysteme". In: *Fahrerassistenzsysteme mit maschineller Wahrnehmung.* Hrsg. von M. Maurer und C. Stiller. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 59–88 (siehe S. 21 f.).

Doering, N., K. Richter, H.-M. Gross, Ch. Schroeter, St. Mueller, M. Volkhardt u. a. (2015). "Robotic Companions for Older People: A Case Study in the Wild". In: *Studies in Health Technology and Informatics* 219, S. 147–152 (siehe S. 123).

Dollár, P., Z. Tu, P. Perona und S. Belongie (2009). "Integral channel features". In: *British machine vision ...* S. 1–11 (siehe S. 37).

Dollár, P., R. Appel und W. Kienzle (2012a). "Crosstalk cascades for frame-rate pedestrian detection". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).* Bd. 7573 LNCS, S. 645–659 (siehe S. 38).

Dollar, P., S. Belongie und P. Perona (2010). "The Fastest Pedestrian Detector in the West". In: *Proceedings of the British Machine Vision Conference.* doi:10.5244/C.24.68. BMVA Press, S. 68.1–68.11 (siehe S. 31 f., 37 f., 51, 76).

Dollár, P., C. Wojek, B. Schiele und P. Perona (2012b). "Pedestrian Detection: An Evaluation of the State of the Art". In: *PAMI* 34 (siehe S. 15, 32).

DOMEO (2009). *Webseite*. `www.aal-domeo.eu` (siehe S. 17).

Double (2013). *Double Robotics Webseite*. `www.doublerobotics.com` (siehe S. 16).

Dubout, C. und F. Fleuret (2012). "Exact Acceleration of Linear Object Detectors." In: *ECCV (3)*. Hrsg. von A. W. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato und C. Schmid. Bd. 7574. Lecture Notes in Computer Science. Springer, S. 301–311 (siehe S. 37, 52, 90).

Einhorn, E. und H.-M. Gross (2014). "Generic NDT mapping in dynamic environments and its application for lifelong SLAM." In: *Robotics and Autonomous Systems* (siehe S. 42, 49).

Einhorn, E., R. Stricker, H.-M. Gross, T. Langner und C. Martin (2012). "MIRA - Middleware for Robotic Applications". In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2012)*. Vilamoura, Portugal: IEEE, S. 2591–2598 (siehe S. 47 f.).

Einhorn, E. (2018). "Visuelle Umgebungswahrnehmung und Kartierung zur Navigation mobiler Serviceroboter in realen Einsatzumgebungen". Diss. Technische Universität Ilmenau (siehe S. 14).

Eisenbach, M., A. Vorndran, S. Sorge und H.-M. Gross (2015). "User Recognition for Guiding and Following People with a Mobile Robot in a Clinical Environment". In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, S. 3600–3607 (siehe S. 129).

Ess, A., B. Leibe, K. Schindler und L. van Gool (Juni 2008). "A Mobile Vision System for Robust Multi-Person Tracking". In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Press (siehe S. 54).

Everingham, M., L. Van Gool, C. K. I. Williams, J. Winn und A. Zisserman (2009). *The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results*. `http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html` (siehe S. 90).

ExCITE (2010). *Webseite*. `www.oru.se/ExCITE` (siehe S. 16, 18, 133).

Felzenszwalb, P. F., R. B. Girshick, D. McAllester und D. Ramanan (2010). "Object Detection with Discriminatively Trained Part Based Models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.9, S. 1627–1645 (siehe S. 31 f., 35 f., 51, 76, 90, 139, 141).

Felzenszwalb, P. und D. Huttenlocher (Sep. 2012). "Distance Transforms of Sampled Functions". In: *Theory of Computing* 8.19 (siehe S. 102).

Ferrari, V., M. Marin-Jimenez und A. Zisserman (Juni 2008). "Progressive Search Space Reduction for Human Pose Estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Anchorage, USA (siehe S. 34, 51).

Fiolka, T., J. Stückler, D. Klein, D. Schulz und S. Behnke (2012). "SURE: Surface Entropy for Distinctive 3D Features". In: *Spatial Cognition VIII*. Bd. 7463. LNCS. Springer, S. 74–93 (siehe S. 111, 115).

Fischler, M. und R. Bolles (1981). "Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography". In: *Commun. ACM* 24 (6), S. 381–395 (siehe S. 113).

Fortmann, T., Y. Bar-Shalom und M. Scheffe (Juli 1983). "Sonar tracking of multiple targets using joint probabilistic data association". In: *IEEE Journal of Oceanic Engineering* 8.3, S. 173–184 (siehe S. 67).

Freund, Y. und R. E. Schapire (Aug. 1997). "A Decision-theoretic Generalization of On-line Learning and an Application to Boosting". In: *J. Comput. Syst. Sci.* 55.1, S. 119–139 (siehe S. 27, 136).

Galarza, A. und J. Seade (2007). *Introduction to Classical Geometries*. SPRINGER VERLAG NY (siehe S. 48).

Gavrila, D. M. und S. Munder (Juli 2006). "Multi-cue Pedestrian Detection and Tracking from a Moving Vehicle". In: *International Journal of Computer Vision* 73.1, S. 41–59 (siehe S. 31 f., 54).

Gehrig, S., A. Barth, N. Schneider und J. Siegemund (Okt. 2012). "A multi-cue approach for stereo-based object confidence estimation". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, S. 3055–3060 (siehe S. 70).

Giebel, J., D. Gavrila und C. Schnörr (2004). "A bayesian framework for multi-cue 3d object tracking". In: *In Proceedings of European Conference on Computer Vision*. Springer-Verlag, S. 241–252 (siehe S. 54).

Giraff (2010). *Giraff Technologies Webseite*. `www.giraff.org` (siehe S. 16).

GiraffPlus (2012). *Giraff Technologies Webseite*. `http://www.giraffplus.eu/` (siehe S. 14, 16, 133).

GmbH Future-Shape (2010). *SensFloor - Hightech fuer mehr Lebensqualität*. `http://www.future-shape.de/en/technologies/23` (siehe S. 110).

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair u. a. (2014). "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27*. Hrsg. von Z. Ghahramani, M. Welling, C.

Cortes, N. D. Lawrence und K. Q. Weinberger. Curran Associates, Inc., S. 2672–2680 (siehe S. 39).

Graf, B. und H. Staab (2009). "Service Robots and Automation for the Disabled/Limited". In: *Handbook of Automation*. Springer, S. 1485–1502 (siehe S. 17).

Gross, H.-M., St. Mueller, Ch. Schroeter, M. Volkhardt, A. Scheidig, K. Debes u. a. (2015). "Robot Companion for Domestic Health Assistance: Implementation, Test and Case Study under Everyday Conditions in Private Apartments". In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, S. 5992–5999 (siehe S. 9 f., 16, 81 f., 106, 123 ff., 130).

Gross, H.-M., Ch. Schroeter, St. Mueller, M. Volkhardt, E. Einhorn, A. Bley u. a. (2011a). "I'll Keep an Eye on You: Home Robot Companion for Elderly People with Cognitive Impairment". In: *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*. IEEE, S. 2481–2488 (siehe S. 11).

– (2011b). "Progress in Developing a Socially Assistive Mobile Home Robot Companion for the Elderly with Mild Cognitive Impairment". In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, S. 2430–2437 (siehe S. 11).

Gross, H.-M., Ch. Schroeter, St. Mueller, M. Volkhardt, E. Einhorn, A. Bley u. a. (2012). "Further Progress towards a Home Robot Companion for People with Mild Cognitive Impairment". In: *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*. IEEE, S. 637–644 (siehe S. 3, 10, 16, 131).

Groves, P. D. (2013). *Principles of GNSS, inertial, and multisensor inte- grated navigation systems*. Artech House (siehe S. 68).

Hegger, F., N. Hochgeschwender, K. Kraetzschmar und P. Ploeger (2012). "People Detection in 3d Point Clouds using Local Surface Normals". In: *Proc. of the Robocup Symposium 2012* (siehe S. 111, 113 f.).

Hegger, F., N. Hochgeschwender, G. Kraetzschmar und P. Ploeger (2013). "People Detection in 3d Point Clouds Using Local Surface Normals". In: *RoboCup 2012: Robot Soccer World Cup XVI*. Hrsg. von X. Chen, P. Stone, L. Sucar und T. van der Zant. Bd. 7500. Lecture Notes in Computer Science. Springer Berlin Heidelberg, S. 154–165 (siehe S. 40, 52).

Hoff, W. und T. Vincent (Okt. 2000). "Analysis of head pose accuracy in augmented reality". In: *Visualization and Computer Graphics, IEEE Transactions on* 6.4, S. 319–334 (siehe S. 48 f.).

Holz, D., N. Basilico, F. Amigoni und S. Behnke (Juni 2010). "Evaluating the Efficiency of Frontier-based Exploration Strategies". In: *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, S. 1–8 (siehe S. 97).

Hordern, D. und N. Kirchner (2010). "Robust and Efficient People Detection with 3-D Range Data using Shape Matching". In: *Proc. of the 2010 Aust. Conf. on Robotics and Automation*, S. 1–8 (siehe S. 40, 52).

Horevych, D. (März 2014). "Verbesserte Personendetektion in den Tiefendaten der Kinect". Masterarbeit. TU Ilmenau (siehe S. 40).

Hosang, J., R. Benenson und B. Schiele (2014). "How good are detection proposals, really?" In: *BMVC* (siehe S. 37).

Huijnen, C., A. Badii und H. van den Heuvel (2011). "Maybe it becomes a buddy, but do not call it a robot - seamless cooperation between companion robotics and smart homes". In: *Ambient Intelligence*, S. 324–329 (siehe S. 1, 17, 109).

Ikemura, S. und H. Fujiyoshi (2010a). "Real-Time Human Detection using Relational Depth Similarity Features". In: *Proc. of the 10th Asian Conf. on Computer Vision*, S. 25–38 (siehe S. 111).

Ikemura, S. und H. Fujiyoshi (2010b). "Real-Time Human Detection using Relational Depth Similarity Features". In: *Proc. of the 10th Asian Conf. on Computer Vision*, S. 25–38 (siehe S. 15, 40).

Isard, M. und A. Blake (1998). "CONDENSATION – conditional density propagation for visual tracking". In: *International Journal of Computer Vision*. Bd. 29. 1, S. 5–28 (siehe S. 56, 62).

Jaakkola, H. und B. Thalheim (2011). "Architecture-driven modelling methodologies". In: *Proceedings of the 2011 conference on Information Modelling and Knowledge Bases XXII*. Anneli Heimbürger et al., IOS Press, S. 98 (siehe S. 18).

Jafari, O., D. Mitzel und B. Leibe (Mai 2014). "Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras". In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, S. 5636–5643 (siehe S. 40, 51 f., 55).

Jayawardena, C., I. H. Kuo, U. Unger, A. Igic, R. Wong, C. Stafford u. a. (2010). "Deployment of a Service Robot to Help Older People". In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2010)*, S. 5990–5995 (siehe S. 17).

Jazz (2011). *Gostai. Webseite*. `www.gostai.com` (siehe S. 16).

Julier, S. J. und J. K. Uhlmann (2001). "General Decentralized Data Fusion with Covariance Intersection". In: Hrsg. von D. L. Hall und J. Llinas. Handbook of Multisensor Data Fusion. Boca Raton, FL: CRC Press (siehe S. 67).

Julier, S. J., J. K. Uhlmann und H. F. Durrant-Whyte (März 2000). "A new method for the nonlinear transformation of means and covariances in filters and estimators". In: *IEEE Transactions on Automatic Control* 45.4, S. 477–482 (siehe S. 56, 61).

Julier, S. und J. Uhlmann (Juni 1997). "A non-divergent estimation algorithm in the presence of unknown correlations". In: *American Control Conference, 1997. Proceedings of the 1997*. Bd. 4, 2369–2373 vol.4 (siehe S. 67).

Julier, S., J. Uhlmann und H. Durrant-Whyte (Juni 1995). "A new approach for filtering nonlinear systems". In: *American Control Conference, Proceedings of the 1995*. Bd. 3, 1628–1632 vol.3 (siehe S. 60 f.).

Julier, S. J. und J. K. Uhlmann (1997). "A New Extension of the Kalman Filter to Nonlinear Systems". In: *International Symposium on Aerospace/Defense Sensing, Simulate and Controls*. Orlando, Florida, S. 182–193 (siehe S. 60).

Kaempchen, N. und K. Dietmayer (2003). "Data Synchronization Strategies for Multi-Sensor Fusion". In: *In Proceedings of the IEEE Conference on Intelligent Transportation Systems*, S. 1–9 (siehe S. 68).

Kaestner, R., J. Maye und R. Siegwart (Mai 2012). "Generative Object Detection and Tracking in 3D Range Data". In: *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)* (siehe S. 55).

Kalman, R. E. (1960). "A New Approach to Linear Filtering and Prediction Problems". In: *Transaction of the ASME, Journal of Basic Engineering* 82.1, S. 35–45 (siehe S. 56, 58).

Kampai (2009). *Robosoft. Webseite*. `http://www.robosoft.com/` (siehe S. 17).

Karpathy, A. und L. Fei-Fei (2017). "Deep Visual-Semantic Alignments for Generating Image Descriptions". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 39.4, S. 664–676 (siehe S. 31, 39).

Kennedy, J. und R. Eberhart (Nov. 1995). "Particle swarm optimization". In: *Neural Networks, 1995. Proceedings., IEEE International Conference on*. Bd. 4, 1942–1948 vol.4 (siehe S. 99).

Kim, C., F. Li, A. Ciptadi und J. M. Rehg (2015). "Multiple Hypothesis Tracking Revisited". In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. ICCV '15. Washington, DC, USA: IEEE Computer Society, S. 4696–4704 (siehe S. 67).

Klein, D. A., D. Schulz, S. Frintrop und A. B. Cremers (2010). "Adaptive Real-Time Video-Tracking for Arbitrary Objects". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan*, S. 772–777 (siehe S. 55).

Kleinehagenbrock, M., S. Lang, J. Fritsch, F. Lömker, G. A. Fink und G. Sagerer (Sep. 2002). "Person Tracking with a Mobile Robot based on Multi-Modal Anchoring". In: *Proc. IEEE Int. Workshop on Robot and Human Interactive Communication (ROMAN)*. Berlin, Germany: IEEE, S. 423–429 (siehe S. 26).

Kondaxakis, P., H. Baltzakis und P. Trahanias (Okt. 2009). "Learning moving objects in a multi-target tracking scenario for mobile robots that use laser range measurements". In: *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, S. 1667–1672 (siehe S. 27).

König, A., S. Müller und H.-M. Gross (2005). "Appearance-based MCL Approach for a Home Store Environment." In: *Proc. of the 2nd European Conference on Mobile Robots (ECMR 2005)*. Ancona, Italy, S. 206–211 (siehe S. 49).

Konstantinova, P., A. Udvarev und T. Semerdjiev (2003). "A study of a target tracking algorithm using global nearest neighbor approach". In: *Proceedings of the 4th international conference conference on Computer systems and technologies e-Learning - CompSysTech '03*, S. 290–295 (siehe S. 66).

Kristoffersson, A., S. Coradeschi, K. S. Eklundh und A. Loutfi (2011). "Sense of Presence in a Robotic Telepresence Domain". In: *Proc. Int. Conf. on Human-Computer Interaction (HCI 2011)*, S. 479–487 (siehe S. 16).

Krizhevsky, A., I. Sutskever und G. E. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Hrsg. von F. Pereira, C. J. C. Burges, L. Bottou und K. Q. Weinberger. Curran Associates, Inc., S. 1097–1105 (siehe S. 39).

KSERA (2010). *Webseite*. `ksera.ieis.tue.nl` (siehe S. 17).

Kubertschak, T., M. Maehlisch und H. J. Wuensche (Juli 2014). "Towards a unified architecture for mapping static environments". In: *17th International Conference on Information Fusion (FUSION)*, S. 1–8 (siehe S. 21).

Kublbeck, C. und A. Ernst (Juni 2006). "Face detection and tracking in video sequences using the modifiedcensus transformation". In: *Image and Vision Computing* 24.6, S. 564–572 (siehe S. 75).

Kuhn, H. W. und B. Yaw (1955). "The Hungarian method for the assignment problem". In: *Naval Res. Logist. Quart*, S. 83–97 (siehe S. 66).

Laschka, A. (2013). "Robuste körperteilbasierte visuelle Personendetektion unter Verdeckungen". Masterarbeit. TU Ilmenau (siehe S. 33 f., 37, 90, 139).

Lau, B., K. Arras und W. Burgard (Mai 2009). "Tracking groups of people with a multi-model hypothesis tracker". In: *2009 IEEE International Conference on Robotics and Automation*. IEEE, S. 3180–3185 (siehe S. 26, 56).

Lee, J. H., K. Abe, T. Tsubouchi, R. Ichinose, Y. Hosoda und O. Ohba (Sep. 2008). "Collision-free navigation based on people tracking algorithm with biped walking model". In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, S. 2983–2989 (siehe S. 55).

Leibe, B. und B. Schiele (Sep. 2003). "Interleaved Object Categorization and Segmentation". In: *British Machine Vision Conference (BMVC'03)*. Norwich, UK (siehe S. 31 f., 54).

Leibe, B. (2005). "Interleaved object categorization and segmentation". Diss. ETH Zurich (siehe S. 158).

Leibe, B., K. Schindler, N. Cornelis und L. Van Gool (Okt. 2008). "Coupled object detection and tracking from static cameras and moving vehicles." In: *IEEE transactions on pattern analysis and machine intelligence* 30.10, S. 1683–98 (siehe S. 54 f., 122).

Lerner, R. und G. Trigg (1991). *Encyclopaedia of Physics*. 2. Aufl. ISBN (Verlagsgesellschaft) 3-527-26954-1 (VHC Inc.) 0-89573-752-3. VHC Publishers (siehe S. 149).

Lewandowski, B., T. Wengefeld, T. Schmiedel und H.-M. Gross (2017). "I See You Lying on the Ground - Can I Help You? Fast Fallen Person Detection in 3D with a Mobile Robot." In: *IEEE Int. Symp. on Robot and Human Interactive Communication (RO-MAN)*. Lisbon, Portugal: IEEE, S. 74–80 (siehe S. 111, 122, 129).

Lewandowski, B. (2016). "Robust Depth Data Based Fallen Person Detection in Domestic Environments". Masterarbeit. TU Ilmenau (siehe S. 122).

Liu, H. und H. He (Mai 2010). "A salient feature and scene semantics based attention model for human tracking on mobile robots". In: *2010 IEEE In-*

ternational Conference on Robotics and Automation. IEEE, S. 4545–4552 (siehe S. 55).

Lord, S. R., C. Sherringtion und H. B. Menz (2003). "Falls in Older People: Risk Factors and Strategies for Prevention". In: *Injury Prevention* 9.1, S. 93–94 (siehe S. 1, 109).

Lowe, D. G. (2004). "Distinctive image features from scale-invariant keypoints". In: *International Journal of Computer Vision* 60.2, S. 91–110 (siehe S. 33).

Luber, M., G. Tipaldi und K. Arras (2011). "Better Models For People Tracking". In: *ICRA*, S. 854–859 (siehe S. 56).

Luna (2014). *RoboDynamics/Luna Webseite* (siehe S. 17).

Luo, R., Y. Chen, C. Liao und A. Tsai (Dez. 2007). "Mobile robot based human detection and tracking using range and intensity data fusion". In: *Advanced Robotics and Its Social Impacts, 2007. ARSO 2007. IEEE Workshop on*, S. 1–6 (siehe S. 26).

Lv, Q. (2011). "A Poselet-based Approach for Fall Detection". In: *Int. Symposium on IT in Medicine and Education*, S. 209–212 (siehe S. 111).

Madrigal, F. und J.-B. Hayet (Aug. 2013). "Evaluation of multiple motion models for multiple pedestrian visual tracking". In: *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance* 1, S. 31–36 (siehe S. 65).

Maehlisch, M., W. Ritter und K. Dietmayer (Juni 2007). "De-cluttering with Integrated Probabilistic Data Association for Multisensor Multitarget ACC Vehicle Tracking". In: *2007 IEEE Intelligent Vehicles Symposium*, S. 178–183 (siehe S. 70).

Martin, C., E. Schaffernicht, A. Scheidig und H.-M. Gross (2006). "Sensor Fusion using a Probabilistic Aggregation Scheme for People Detection and People Tracking." In: *Robotics and Autonomous Systems* 54.9, S. 721–728 (siehe S. 25).

Mastorakis, G. und D. Makris (2012). "Fall detection system using Kintect's infrared sensor". In: *Real-Time Image Processing* (siehe S. 111).

Mathias, M., R. Benenson, M. Pedersoli und L. Van Gool (2014). "Face detection without bells and whistles". In: *ECCV* (siehe S. 76).

Meyer, S. (2011). *Mein Freund der Roboter*. VDE Verlag GmbH (siehe S. 1).

Mitzel, D. und B. Leibe (2012). "Close-Range Human Detection for Head-Mounted Cameras". In: *British Machine Vision Conference (BMVC'12)* (siehe S. 40).

Mitzel, D., E. Horbert, A. Ess und B. Leibe (2010). "Multi-person Tracking with Sparse Detection and Continuous Segmentation". In: *Computer Vision–ECCV 2010, Lecture Notes in Computer Science* 6311/2010, S. 397–410 (siehe S. 56).

Mitzel, D., P. Sudowe und B. Leibe (2011). "Real-Time Multi-Person Tracking with Time-Constrained Detection". In: *Proceedings of the British Machine Vision Conference*. BMVA Press, S. 104.1–104.11 (siehe S. 54).

Mobiserv (2009). *Webseite*. `http://www.mobiserv.info/` (siehe S. 17 f., 134).

Montemerlo, M., J. Pineau, N. Roy, S. Thrun und V. Verma (2002). "Experiences with a Mobile Robotic Guide for the Elderly." In: *AAAI/IAAI*. Hrsg. von R. Dechter und R. S. Sutton. AAAI Press / The MIT Press, S. 587–592 (siehe S. 16).

Mozos, O. M., R. Kurazume und H. Tsutomu (Jan. 2010). "Multi-Part People Detection Using 2D Range Data". In: *International Journal of Social Robotics* 2.1, S. 31–40 (siehe S. 15, 26, 51).

Mucientes, M. und W. Burgard (Okt. 2006). "Multiple Hypothesis Tracking of Clusters of People". In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, S. 692–697 (siehe S. 26, 56).

Müller, S., E. Schaffernicht, A. Scheidig, H.-J. Böhme und H.-M. Gross (2007). "Are you still following me?" In: *Proc. 3rd European Conference on Mobile Robots (ECMR)*. Freiburg: Albert-Ludwigs-Universität Freiburg - Universitätsverlag, S. 211–216 (siehe S. 25 f.).

Munaro, M., F. Basso und E. Menegatti (Okt. 2012). "Tracking people within groups with RGB-D data". In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, S. 2101–2107 (siehe S. 39).

Munaro, M. und E. Menegatti (Okt. 2014). "Fast RGB-D People Tracking for Service Robots". In: *Auton. Robots* 37.3, S. 227–242 (siehe S. 39).

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: The MIT Press (siehe S. 58, 60, 62).

Musicki, D., R. Evans und S. Stankovic (Juni 1994). "Integrated probabilistic data association". In: *IEEE Transactions on Automatic Control* 39.6, S. 1237–1241 (siehe S. 67).

Navarro-Serment, L. E., C. Mertz und M. Hebert (Okt. 2010). "Pedestrian Detection and Tracking Using Three-dimensional LADAR Data". In: *The International Journal of Robotics Research, Special Issue on the Seventh International Conference on Field and Service Robots* 29.12, S. 1516–1528 (siehe S. 26, 40).

Noury, N., P. Rumeau, A. Bourke, G. Laighin und J. Lundy (2008). "A proposal for the classification and evaluation of fall detectors". In: *IRBM* 29.6, S. 340–349 (siehe S. 109).

Padbot (2014). *Webseite*. `http://www.padbot.co/` (siehe S. 16).

Pantofaru, C. (2011). "User observation & dataset collection for robot training". In: *Human-Robot Interaction (HRI), 2011 6th ACM/IEEE International Conference on*, S. 217–218 (siehe S. 55).

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel u. a. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, S. 2825–2830 (siehe S. 28).

Philippsen, R. und R. Siegwart (Apr. 2005). "An Interpolated Dynamic Navigation Function". In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, S. 3782–3789 (siehe S. 101).

Philips Electronics (2012). *AutoAlert Pendant*. `http://www.lifelinesys.com/content/lifeline-products/personal-help-buttons/auto-alert-pendant` (siehe S. 110).

Plagemann, C. (2010). "Real-time identification and localization of body parts from depth images". In: *IEEE ICRA*, S. 3108–3113 (siehe S. 111).

Plagemann, C., V. Ganapathi, D. Koller und S. Thrun (Mai 2010). "Real-time identification and localization of body parts from depth images". In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, S. 3108–3113 (siehe S. 40).

Planinc, R. und M. Kampel (2012). "Introducing the use of depth data for fall detection". In: *Personal and Ubiquitous Computing*, S. 1–10 (siehe S. 110).

Popescu, M. und A. Mahnot (2009). "Acoustic fall detection using one-class classifier". In: *Annual Int. Conf of the IEEE Engineering in Medicine and Biology Society*, S. 3505–3508 (siehe S. 110).

PR2 (2010). *Willow Garage/PR2 Webseite*. `http://www.willowgarage.com/pages/pr2/overview` (siehe S. 17 f., 82, 134).

Premebida, C., O. Ludwig und U. Nunes (Okt. 2009). "Exploiting LIDAR-based features on pedestrian detection in urban scenarios". In: *Intelligent*

Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on, S. 1–6 (siehe S. 27).

Premebida, C. und U. Nunes (2005). "Segmentation and geometric primitives extraction from 2d laser range data for mobile robot applications". In: *Robotica*, S. 17–25 (siehe S. 26).

QB (2010). *Anybots. Webseite*. `www.anybots.com` (siehe S. 16).

Ramanan, D., D. Forsyth und A. Zisserman (Juni 2005). "Strike a pose: tracking people by finding stylized poses". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Bd. 1, 271–278 vol. 1 (siehe S. 32).

Ramanan, D. (2012). "Face Detection, Pose Estimation, and Landmark Localization in the Wild". In: *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. CVPR '12. Washington, DC, USA: IEEE Computer Society, S. 2879–2886 (siehe S. 31).

Redmon, J., S. K. Divvala, R. B. Girshick und A. Farhadi (2015). "You Only Look Once: Unified, Real-Time Object Detection". In: *CoRR* abs/1506.02640 (siehe S. 31, 39).

Redmon, J. und A. Farhadi (2016). "YOLO9000: Better, Faster, Stronger". In: *CoRR* abs/1612.08242 (siehe S. 39).

Reid, D. B. (1979). "An Algorithm for Tracking Multiple Targets". In: *IEEE Transactions on Automatic Control* 24, S. 843–854 (siehe S. 66 f.).

Ren, S., K. He, R. Girshick und J. Sun (2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in Neural Information Processing Systems (NIPS)* (siehe S. 39).

Reuther, C. (2011). "Aufbereitung eines Verfahrens zur körperteilbasierten Personendetektion in Gradientenbildern". Hauptseminar. TU Ilmenau (siehe S. 36, 139).

Rheaume, F. und A. R. Benaskeur (Dez. 2008). "Forward prediction-based approach to target-tracking with Out-of-Sequence Measurements". In: *2008 47th IEEE Conference on Decision and Control*, S. 1326–1333 (siehe S. 69).

Richter, E. (2012). *Non-Parametric Bayesian Filtering for Multiple Object Tracking*. Forschungsberichte der Professur Nachrichtentechnik. Shaker Verlag (siehe S. 57, 129).

Rijsbergen, C. J. V. (1979). *Information Retrieval*. 2nd. Newton, MA, USA: Butterworth-Heinemann (siehe S. 116, 157).

Robinson, H., B. A. MacDonald und E. Broadbent (2014). "The Role of Healthcare Robots for Older People at Home: A Review". In: *I. J. Social Robotics* 6.4, S. 575–591 (siehe S. 2, 16).

Romeo 2 (2012). *Webseite*. `http://projetromeo.com/` (siehe S. 17).

ROREAS (2013). *Interaktiver robotischer Reha-Assistent für das Lauf- und Orientierungstraining von Patienten nach Schlaganfällen*. `www.tu-ilmenau.de/neurob/projects/roreas/` (siehe S. 3).

Rother, C., V. Kolmogorov und A. Blake (Aug. 2004). "GrabCut: Interactive Foreground Extraction Using Iterated Graph Cuts". In: *ACM Trans. Graph.* 23.3, S. 309–314 (siehe S. 86).

Rougier, C., E. Auvinet und J. Rousseau (Juni 2011). "Fall Detection from Depth Map Video Sequences". In: *Towards Useful Services for Elderly an People with Disabilities: 9th Int. Conf. on Smart Homes and Health Telematics (ICOST'11)*. Montreal, Kanada, S. 121–128 (siehe S. 111).

Rubinstein, R. Y. und D. P. Kroese (2008). *Simulation and the Monte Carlo Method (Wiley Series in Probability and Statistics)*. 2. Aufl. (siehe S. 56, 62).

Rusu, R. (2009). "Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments". Diss. Computer Science department, Technical University Munich (siehe S. 111, 115).

Saint-Aime, S., B. Le-Pevedic, D. Duhaut und T. Shibata (2007). "EmotiRob: Companion Robot Project". In: *Proc. IEEE Intl. Conf. on Robot & Human Interactive Communication (Ro-Man 2007)*. Jeju, Korea, S. 919–924 (siehe S. 17).

Scheidig, A., K. Debes, St. Mueller, Ch. Schroeter, M. Volkhardt, H.-M. Gross u. a. (2015). "SERROGA: Funktions- und Nutzertests Herangehensweise und Ergebnisse". In: *German AAL Conference (AAL)*. VDE, S. 34–43 (siehe S. 9).

Scheidig, A., Ch. Schroeter, M. Volkhardt, St. Mueller, K. Debes, H.-M. Gross u. a. (2014). "SERROGA: Servicerobotik fuer die Gesundheitsassistenz im nutzerzentrierten Entwurf". In: *German AAL Conference (AAL)*. VDE (siehe S. 9).

Schenk, K., M. Eisenbach, A. Kolarow und H.-M. Gross (2011). "Comparison of Laser-based Person Tracking at Feet and Upper-Body Height." In: *Proc. 34th Annual Conference on Artificial Intelligence (KI 2011)*. Hrsg. von L. 7006. Berlin, Germany: Springer 201, S. 277–288 (siehe S. 25 f.).

Schmiedel, Th., E. Einhorn und H.-M. Gross (2015). "IRON: A Fast Interest Point Descriptor for Robust NDT-Map Matching and Its Application to Robot Localization". In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, S. 3144–3151 (siehe S. 111).

Schneemann, F. (2013). "Recherche und Evaluation von Features zur Detektion von gestürzten Personen in häuslichen Umgebungen". Masterarbeit. TU Ilmenau (siehe S. 109 f., 115).

Schroeter, Ch., St. Mueller, M. Volkhardt, E. Einhorn, H.-M. Gross, C. Huijnen u. a. (2014). "CompanionAble – Ein robotischer Assistent und Begleiter fuer Menschen mit leichter kognitiver Beeintraechtigung". In: *German AAL Conference (AAL)* (siehe S. 9).

Schröter, C., S. Müller, M. Volkhardt, E. Einhorn, C. Huijnen, H. van den Heuvel u. a. (2013). "Realization and User Evaluation of a Companion Robot for People with Mild Cognitive Impairments." In: *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA 2013)*. IEEE, S. 1145–1151 (siehe S. 10, 16).

Schubert, R. (2011). *Integrated Bayesian Object and Situation Assessment for Lane Change Assistance*. Forschungsberichte der Professur Nachrichtentechnik. Shaker (siehe S. 57, 129).

Schubert, R., E. Richter, N. Mattern, P. Lindner und G. Wanielik (2010). "A Concept Vehicle for Rapid Prototyping of Advanced Driver Assistance Systems". In: *Advanced Microsystems for Automotive Applications 2010: Smart Systems for Green Cars and Safe Mobility*. Hrsg. von G. Meyer und J. Valldorf. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 211–219 (siehe S. 21, 44).

Schulz, D., W. Burgard, D. Fox und A. Cremers (2003). "People Tracking with Mobile Robots Using Sample-based Joint Probabilistic Data Association Filters". In: *International Journal of Robotics Research* 22.2 (siehe S. 25 f.).

Schwarz, L. (2011). "Estimating human 3D pose from Time-of-Flight images based on geodesic distances and optical flow". In: *Int. Conf. and Workshop on Automatic Face & Gesture Recognition*, S. 700–706 (siehe S. 111).

SERROGA (2012). *Service-Robotik für die Gesundheitsassistenz*. `www.serroga.de` (siehe S. 2, 16, 18, 106, 123, 132).

Shao, X., K. Katabira, R. Shibasaki und H (Okt. 2008). "Tracking a variable number of pedestrians in crowded scenes by using laser range scanners".

In: *2008 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, S. 1545–1551 (siehe S. 26).

Shoaib, M., R. Dragon und J. Ostermann (2011). "Context-aware visual analysis of elderly activity in a cluttered home environment". In: *EURASIP Journal on Advances in Signal Processing*, S. 1–14 (siehe S. 110).

Simonyan, K. und A. Zisserman (2014). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556 (siehe S. 39).

Simpson, A. L., B. Ma, R. E. Ellis, A. J. Stewart und M. I. Miga (März 2011). "Uncertainty propagation and analysis of image-guided surgery". In: *Proc. SPIE 7964, Medical Imaging 2011: Visualization, Image-Guided Procedures, and Modeling*. Bd. 7964 (siehe S. 49).

Smeulders, A., D. Chu, R. Cucchiara, S. Calderara, A. Dehghan und M. Shah (Juli 2014). "Visual Tracking: An Experimental Survey". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 36.7, S. 1442–1468 (siehe S. 55).

Spinello, L., M. Luber und K. O. Arras (2011a). "Tracking People in 3D Using a Bottom-Up Top-Down Detector." In: *Proc. of The International Conference in Robotics and Automation (ICRA)* (siehe S. 26, 40, 51).

Spinello, L., R. Triebel und R. Siegwart (Okt. 2010a). "Multiclass Multimodal Detection and Tracking in Urban Environments". In: *The International Journal of Robotics Research* 29.12, S. 1498–1515 (siehe S. 54, 65).

Spinello, L., K. O. Arras, R. Triebel und R. Siegwart (2010b). "A Layered Approach to People Detection in 3D Range Data". In: *Proc. of the 24th AAAI Conf. on Artificial Intelligence* (siehe S. 15, 27, 111, 113 f.).

Spinello, L. und K. O. Arras (Sep. 2011b). "People detection in RGB-D data". In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, S. 3838–3843 (siehe S. 40 f., 52, 111).

Spinello, L. und K. O. Arras (2012). "Leveraging RGB-D Data: Adaptive fusion and domain adaptation for object detection." In: *ICRA*. IEEE, S. 4469–4474 (siehe S. 20).

Stauffer, C. und W. E. L. Grimson (Aug. 1999). "Adaptive background mixture models for real-time tracking". In: *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*. Bd. 2. Artificial Intelligence Lab., MIT, Cambridge, MA, USA. Los Alamitos, CA, USA: IEEE, S. 246–252 (siehe S. 25, 86).

Steder, B., R. Rusu, K. Konolige und W. Burgard (2010). "NARF: 3D range image features for object recognition". In: *Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at IROS* (siehe S. 111).

Stiebritz, K. (2014). "Analyse von Bewegungsmustern zur Bestimmung der körperlichen und geistigen Fitness". Masterarbeit. TU Ilmenau (siehe S. 83).

Stricker, R., St. Mueller, E. Einhorn, Ch. Schroeter, M. Volkhardt, K. Debes u. a. (2012a). "Interactive Mobile Robots Guiding Visitors in a University Building". In: *IEEE Int. Symp. on Robot and Human Interactive Communication (RO-MAN)*. IEEE, S. 695–700 (siehe S. 11).

– (2012b). "Konrad and Suse, Two Robots Guiding Visitors in a University Building". In: *Autonomous Mobile Systems 2012 (AMS)*. Informatik aktuell. Springer, S. 49–58 (siehe S. 11).

Stückler, J. und S. Behnke (2011). "Improving People Awareness of Service Robots by Semantic Scene Knowledge". In: *RoboCup 2010: Robot Soccer World Cup XIV*. Hrsg. von J. Ruiz-del-Solar, E. Chown und P. G. Plöger. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 157–168 (siehe S. 97).

Suck, S. (März 2013). "Personendetektion auf einem mobilen Roboter mittels Kinect". Diplomarbeit. TU Ilmenau (siehe S. 40).

Sudowe, P. und B. Leibe (2011). "Efficient use of geometric constraints for sliding-window object detection in video". In: *Proceedings of the 8th international conference on Computer vision systems*, S. 11–20 (siehe S. 35).

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov u. a. (2015). "Going Deeper with Convolutions". In: *Computer Vision and Pattern Recognition (CVPR)* (siehe S. 39).

Tang, S. (2011). *Visual recognition using hybrid cameras*. Thesis, University of Missouri, USA (siehe S. 111).

Tapus, A., M. J. Matarić und B. Scassellati (2007). "The Grand Challenges in Socially Assistive Robotics". In: *IEEE Robotics and Automation Magazine* 14.1, S. 35–42 (siehe S. 3, 16, 131).

Texai (2012). *Willow Garage/Texai Webseite*. `www.willowgarage.com/pages/texai` (siehe S. 16).

Thrun, S., W. Burgard und D. Fox (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press (siehe S. 53, 58, 60 ff., 97 ff., 143–147, 164).

Treptow, A., G. Cielniak und T. Duckett (Sep. 2005). "Comparing Measurement Models for Tracking People in Thermal Images on a Mobile Robot".

In: *Proc. 2nd European Conference on Mobile Robots (ECMR)*. Ancona, Italy (siehe S. 31).

Uhlmann, J. K. (2001). "Introduciton to the algorithmics of data association in multiple-target tracking". In: *Handbook of Multisensor Data Fusion*, S. 3.1–3.17 (siehe S. 65 f.).

Vaidehi, A., K. Ganapathy, K. Mohan, A. Aldrin und K. Nirmal (2011). "Video based automatic fall detection in indoor environment". In: *Int. Conf. on Recent Trends in Information Technology*, S. 1016–1020 (siehe S. 110).

Valencia, R., J. V. Miró, G. Dissanayake und J. Andrade-Cetto (Okt. 2012). "Active Pose SLAM". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, S. 1885–1891 (siehe S. 97).

Vallvé, J. und J. Andrade-Cetto (2015). "Potential information fields for mobile robot exploration". In: *Robotics and Autonomous Systems* 69. Selected papers from 6th European Conference on Mobile Robots, S. 68–79 (siehe S. 97).

Ven, A., A. Sponselee und B. Schouten (2010). "Robo M.D.: A Home Care Robot for Monitoring and Detection of Critical Situations". In: *Europ. Conf. on Cognitive Ergonomics (ECCE 2010)*. Delft, The Netherlands, S. 375–376 (siehe S. 17).

VGo (2011). *VGo Communications company Webseite*. `www.vgocom.com` (siehe S. 16, 18, 133).

Viola, P. und M. Jones (2002). "Robust real-time object detection". In: *International Journal of Computer Vision* 57.2, S. 137–154 (siehe S. 27, 31 ff., 51, 98, 136).

Volkhardt, M., C. Weinrich und H.-M. Gross (2013a). "People Tracking on a Mobile Companion Robot". In: *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics (IEEE-SMC 2013)*. Manchester, GB: IEEE Computer Society CPS, S. 4354–4359 (siehe S. 7, 34, 56, 63, 72 f., 75 f., 79 f., 128, 151, 158).

Volkhardt, M. (Nov. 2008). "Integrativer probabilistischer Personentracker für die persistente Nutzermodellierung". Diplomarbeit. TU Ilmenau (siehe S. 129).

Volkhardt, M. und H.-M. Gross (2013b). "Finding People in Apartments with a Mobile Robot". In: *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*. IEEE, S. 4348–4353 (siehe S. 8, 88, 90, 92, 96, 98).

– (2013c). "Finding People in Home Environments with a Mobile Robot". In: *Europ. Conf. on Mobile Robots (ECMR)* (siehe S. 8, 88, 90, 96).

Volkhardt, M., S. Kalesse, St. Mueller und H.-M. Gross (2009a). "Maximum a Posteriori Estimation of Dynamically Changing Distributions". In: *German Conf. on Artificial Intelligence (KI)*. Bd. 5803. LNAI. Springer, S. 484–491 (siehe S. 11).

Volkhardt, M., St. Mueller, Ch. Schroeter und H.-M. Gross (2010). "Real-Time Activity Recognition on a Mobile Companion Robot". In: *Int. Scientific Colloquium Ilmenau (IWK)*. ISLE Verlag, S. 612–617 (siehe S. 11, 83).

– (2011a). "Detection of Lounging People with a Mobile Robot Companion". In: *Int. Conf. on Intelligent Robotics and Applications (ICIRA)*. Bd. 7102. LNCS 2. Springer, S. 328–337 (siehe S. 8, 85, 87 f.).

– (2011b). "Playing Hide and Seek with a Mobile Companion Robot". In: *IEEE-RAS Int. Conf. on Humanoid Robots (HUMANOIDS)*. IEEE, S. 40–46 (siehe S. 8, 85 f., 88).

Volkhardt, M., F. Schneemann und H.-M. Gross (2013d). "Fallen Person Detection for Mobile Robots using 3D Depth Data". In: *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*. IEEE, S. 3573–3578 (siehe S. 8, 109, 112, 117, 119 f., 122).

Volkhardt, M., C. Weinrich und H.-M. Gross (2013e). "Multi-Modal People Tracking on a Mobile Companion Robot". In: *Europ. Conf. on Mobile Robots (ECMR)* (siehe S. 8).

Volkhardt, M., C. Weinrich, C. Schröter und H.-M. Gross (2009b). "A Concept for Detection and Tracking of People in Smart Home Environments with a Mobile Robot". In: *2nd CompanionAble Workshop co-located with the 3rd European Conference on Ambient Intelligence November 18th - 21st*. Salzburg, Austria (siehe S. 8, 32).

Wang, S., S. Zabir und S. Leibe (2011). "Lying Pose Recognition for Elderly Fall Detection". In: *Proceedings of Robotics: Science and Systems* (siehe S. 111).

Weiler, D., F. Roehrbein und J. Eggert (Juni 2009). "Level-set segmentation with contour based object representation". In: *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, S. 3327–3334 (siehe S. 31).

Weinrich, C., C. Vollmer und H.-M. Gross (2012). "Estimation of Human Upper Body Orientation for Mobile Robotics using an SVM Decision Tree on Monocular Images." In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2012)*. Vilamoura, Portugal: IEEE, S. 2147–2152 (siehe S. 44, 82, 128).

Weinrich, C., T. Wengefeld, C. Schröter und H.-M. Gross (2014a). "People Detection and Distinction of their Walking Aids in 2D Laser Range Data based on Generic Distance-Invariant Features." In: *Proc. 23rd IEEE Int. Symposium on Robot and Human Interactive Communication (RO-MAN 2014)*. Edinburgh, UK: IEEE, S. 767–773 (siehe S. 26 f., 51).

Weinrich, C., T. Wengefeld, M. Volkhardt, A. Scheidig und H.-M. Gross (2014b). "Generic Distance-Invariant Features for Detection of People with Walking Aid in 2D Range Data". In: *Proc. 13th Int. Conf. on Intelligent Autonomous Systems (IAS 2014)*. Padua, Italy, S. 12 (siehe S. 10, 30, 51, 135).

Weinrich, C. (2016). "Personenwahrnehmung für eine sozialverträgliche und nutzerzentrierte Roboternavigation in öffentlichen Einsatzumgebungen". Dissertation. TU Ilmenau (siehe S. 14, 44).

Weinrich, C., M. Volkhardt, E. Einhorn und H.-M. Gross (2013a). "Prediction of Human Collision Avoidance Behavior by Lifelong Learning for Socially Compliant Robot Navigation". In: *IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, S. 376–381 (siehe S. 10).

Weinrich, C., M. Volkhardt und H.-M. Gross (2013b). "Appearance-Based 3D Upper-Body Pose Estimation and Person Re-Identification on Mobile Robots". In: *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*. IEEE, S. 4384–4390 (siehe S. 10, 44, 82, 128).

Welch, G. und G. Bishop (1995). *An Introduction to the Kalman Filter*. Techn. Ber. Chapel Hill, NC, USA (siehe S. 60).

Wengefeld, T. (2014). "Personendetektion durch Klassifikation von Laserscans". Masterarbeit. TU Ilmenau, Fachgebiet Neuroinformatik und Kognitive Robotik (siehe S. 29 f., 137, 157).

Wengefeld, T., B. Lewandowski und H.-M. Gross (2016). "Detektion gestürzter Personen in häuslicher Einsatzumgebung". In: *Proceedings Innteract Conference 2016*. Awi&I-Wissenschaft & Praxis (siehe S. 122, 129).

Willems, J., G. Debard, B. Bonroy, V. B. und T. Goedeme (2009). "How to detect human fall in video? An overview". In: *Positioning and context-aware international conference*, S. 6388–6391 (siehe S. 111).

Wu, B. und R. Nevatia (Jan. 2007). "Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors". In: *International Journal of Computer Vision* 75.2, S. 247–266 (siehe S. 55).

Wu, S., S. Yu und W. Chen (Dez. 2011). "An attempt to pedestrian detection in depth images". In: *Intelligent Visual Surveillance (IVS), 2011 Third Chinese Conference on*, S. 97–100 (siehe S. 40, 111).

Xavier, J., M. Pacheco, D. Castro, A. E. B. Ruano und U. Nunes (2005). "Fast Line, Arc/Circle and Leg Detection from Laser Scan Data in a Player Driver." In: *Proc. of the IEEE Int. Conference on Robotics & Automation*. IEEE, S. 3930–3935 (siehe S. 26 f.).

Xia, L. (2011). "Human detection using depth information by Kinect". In: *Computer Society Conf. on Computer Vision and Pattern Recognition Workshops*, S. 15–22 (siehe S. 111).

Xudong, M., C. Hu, D. Xianzhong und K. Qian (Sep. 2008). "Sensor integration for person tracking and following with mobile robot". In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, S. 3254–3259 (siehe S. 55, 82).

Yilmaz, A., O. Javed und M. Shah (Dez. 2006). "Object Tracking: A Survey". In: *ACM Computing Surveys* 38.4 (siehe S. 55).

Zhang, C., Y. Tian und E. Capezuti (2012). "Privacy Preserving Automativ Fall Detection for Using RGBD Camers". In: *Int. Conf. on Computers Helping People with Special Needs*, S. 625–633 (siehe S. 111).

Zhang, C. und Z. Zhang (Juni 2010). *A Survey of Recent Advances in Face Detection*. Techn. Ber. MSR-TR-2010-66. Microsoft Research (siehe S. 31, 75).

Zhao, H., Y. Chen, X. Shao, K. Katabira und R. Shibasaki (Apr. 2007). "Monitoring a populated environment using single-row laser range scanners from a mobile platform". In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. April. IEEE, S. 4739–4745 (siehe S. 26).

Zhou, X., C. Yang und W. Yu (Juni 2012). "Moving Object Detection by Detecting Contiguous Outliers in the Low-Rank Representation." In: *IEEE transactions on pattern analysis and machine intelligence* 35.3, S. 597–610 (siehe S. 25).

Zivkovic, Z. und B. Krose (Okt. 2007). "Part based people detection using 2D range data and images". In: *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, S. 214–219 (siehe S. 27).